

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
INFORMÁTICA E ESTATÍSTICA**

Bernardo de Sousa Valverde

**CEIFADOR - UMA ABORDAGEM PARA GERAÇÃO DE  
RELATÓRIOS DE APOIO À PÓS-GRADUAÇÃO**

Florianópolis

2017



Bernardo de Sousa Valverde

## **CEIFADOR - UMA ABORDAGEM PARA GERAÇÃO DE RELATÓRIOS DE APOIO À PÓS-GRADUAÇÃO**

Trabalho de Conclusão de Curso submetido ao Departamento de Informática e Estatística para a obtenção do Grau de Bacharel em Ciências da Computação.

Orientadora: Profa. Dra. Carina Friedrich Dorneles

Florianópolis

2017

Catálogo na fonte elaborada pela biblioteca da  
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:

<http://www.bu.ufsc.br/design/Catalogacao.html>

Bernardo de Sousa Valverde

## **CEIFADOR - UMA ABORDAGEM PARA GERAÇÃO DE RELATÓRIOS DE APOIO À PÓS-GRADUAÇÃO**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Ciências da Computação”, e aprovado em sua forma final pelo Departamento de Informática e Estatística.

Florianópolis, 30 de outubro 2017.

---

Prof. Dr. Eng. Rafael Luiz Cancian  
Coordenador do Curso

### **Banca Examinadora:**

---

Profa. Dra. Carina Friedriech Dorneles  
Orientadora

---

M.Sc. Angelo Augusto Frozza

---

M.Sc. Rodrigo Gonçalves



## RESUMO

Este trabalho apresenta as etapas de desenvolvimento do Ceifador, um sistema utilizado para gerar relatórios de apoio à gestão da pós-graduação. Estes relatórios podem ser utilizados nos diversos métodos de controle de qualidade dos programas de pós-graduação, inclusive como auxílio para as avaliações oficiais às quais eles são submetidos. As informações que constituem os relatórios estão quase que integralmente disponíveis nos currículos Lattes dos profissionais envolvidos, estes podendo ser extraídos da plataforma Lattes em formato XML. Sendo assim, o Ceifador visa consultar os dados presentes nestes documentos utilizando a linguagem *XQuery* e apresentar os resultados obtidos utilizando uma interface que permite uma fácil visualização das informações, automatizando assim uma atividade anteriormente feita de forma manual pelas coordenadorias dos programas.

**Palavras-chave:** Ceifador. Banco de Dados. Relatórios. Pós-Graduação. Currículo Lattes.





## ABSTRACT

This paper presents the development of Ceifador, a system used to generate reports to help the management of graduation programs. These reports may be used in various quality control methods of graduation programs as an aid to the official evaluations to which they are subjected. The information that constitutes these reports is almost fully available in the curriculum Lattes of the involved professionals, extractable from the Lattes platform in XML format. Ceifador consults the data present in these documents by using XQuery and shows the obtained results in an interface that allows easy visualization of the information, automating an activity previously done manually by the programs' management groups.

**Keywords:** Ceifador. Data Bank. Reports. Graduation. Curriculum Lattes.



## LISTA DE FIGURAS

Figura 1	Fragmento de Currículo Lattes no formato XML.....	22
Figura 2	Consulta <i>XQuery</i> sobre Currículo Lattes em formato XML.....	24
Figura 3	Visão geral do funcionamento do sistema .....	33
Figura 4	Visão inicial do Ceifador mostrando a aba de seleção de consultas.....	39
Figura 5	Aplicação de filtro temporal às consultas.....	40
Figura 6	Código de comunicação com o <i>BaseX</i> utilizando a API XQJ .....	40
Figura 7	Visualização do resultado de uma consulta.....	42
Figura 8	Aba de manipulação do banco de dados.....	43
Figura 9	Consulta <i>XQuery</i> que retorna, para cada professor, a lista e o número total de premiações recebidas.....	45
Figura 10	Resgate de atributo sem valor .....	47
Figura 11	Fragmento do relatório de coautorias entre os professores do PPGCC.....	48
Figura 12	Fragmento do relatório de coautorias entre todos os professores .....	49



## LISTA DE TABELAS

Tabela 1	Comparação entre os trabalhos relacionados . . . . .	30
----------	--	----



## LISTA DE ABREVIATURAS E SIGLAS

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior .....	17
PPGCC	Programa de Pós-Graduação em Ciência da Computação .....	17
UFSC	Universidade Federal de Santa Catarina .....	17
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico .....	21
DTD	<i>Document Type Definition</i> .....	21
XML	<i>eXtensible Markup Language</i> .....	22
W3C	<i>World Wide Web Consortium</i> .....	22
LIFO	<i>Last In, First Out</i> .....	22
SQL	<i>Structured Query Language</i> .....	23
SAPos	Sistema de Apoio à Pós .....	26
SGBD	Sistema Gerenciador de Banco de Dados .....	35
BSD	<i>Berkeley Software Distribution</i> .....	36
JVM	<i>Java Virtual Machine</i> .....	36
XQJ	<i>The XQuery API for Java</i> .....	37
API	<i>Application Programming Interface</i> .....	37
ACID	Atomicidade, Consistência, Isolamento e Durabilidade ..	37
HTML	<i>HyperText Markup Language</i> .....	38
CSS	<i>Cascading Style Sheet</i> .....	38





## SUMÁRIO

<b>1 INTRODUÇÃO</b>	17
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	21
2.1 CURRÍCULO LATTES	21
2.2 XML	22
2.3 <i>XQUERY</i>	23
<b>3 TRABALHOS RELACIONADOS</b>	25
3.1 SAPOS	26
3.2 COBALTO	27
3.3 SIGPOS	27
3.4 SIPEX	28
3.5 SIGA	29
3.6 DISCUSSÃO E ANÁLISE COMPARATIVA	30
<b>4 O CEIFADOR</b>	33
4.1 VISÃO GERAL	33
4.1.1 Aplicações	34
4.2 TECNOLOGIAS UTILIZADAS	35
4.2.1 <i>BaseX</i>	36
4.2.2 <i>Java</i>	36
4.2.3 <b>XQJ</b>	37
4.2.4 <b>HTML</b>	38
4.3 INTERFACE	39
4.4 CONSULTAS	43
4.4.1 Consultas projetadas	43
4.4.2 Desenvolvimento das Consultas	45
4.5 DESAFIOS	47
<b>5 CONCLUSÕES</b>	51
<b>REFERÊNCIAS</b>	53
<b>APÊNDICE A – Consultas Desenvolvidas</b>	57
<b>ANEXO A – Avaliação PPGCC UFSC 2013</b>	71



## 1 INTRODUÇÃO

Qualidade de educação é um fator muito importante na hora da escolha de uma instituição de ensino. A métrica oficial da qualidade dos programas de pós-graduação existentes no Brasil se dá através do controle de qualidade da CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior –, órgão do Ministério da Educação responsável pela avaliação quadrienal dos cursos de pós-graduação *stricto sensu*. É imprescindível que os programas de pós-graduação estejam preparados para estas avaliações, uma vez que aqueles que não recebem uma nota maior ou igual a 3 são impedidos de abrir novas turmas (CAPES, 2015). Portanto, a fim de abrir um espaço de diálogo que permita à CAPES orientar e auxiliar os cursos na resolução de possíveis problemas, garantindo, assim, que estejam melhor preparados para a avaliação quadrienal (SOBRINHO, 2017), é feito um acompanhamento anual (CAPES, 2006) baseado em relatórios de atividades acadêmicas e de aplicação de recursos elaborados pela coordenadoria do programa. Estes dados devem ser fornecidos à CAPES através de um sistema *on-line* denominado Plataforma Sucupira (CAPES, 2014).

Alguns dos critérios utilizados nas avaliações periódicas são referentes ao perfil do corpo docente e suas produções intelectuais, além de dados sobre os discentes e propostas do programa, como pode ser visto na avaliação trienal referente ao ano de 2013 do PPGCC UFSC<sup>1</sup>. A titulação dos professores, assim como sua experiência, diversidade de formação e publicações, com ou sem colegas do programa, são fatores determinantes na atribuição da nota. Essas são informações que podem ser aproveitados também em medidas mais internas de controle de qualidade, como planejamentos estratégicos e avaliações de docentes. Disponíveis nos currículos Lattes de cada professor, esses dados são extraídos manualmente pela coordenadoria do programa de pós-graduação, ano após ano.

O PPGCC UFSC conta, atualmente, com mais de 140 estudantes e mais de 25 professores que atuam em diversas áreas da Ciência da Computação<sup>2</sup>. Como os dados necessários para o controle de qualidade são referentes a todos os docentes envolvidos no programa de pós-graduação, além de outros referentes aos discentes, sua coleta manual constitui tarefa excessivamente trabalhosa para a execução integral da coordenadoria do programa, que deve cumprir também com outras

---

<sup>1</sup>Anexo A

<sup>2</sup><<http://ppgcc.posgrad.ufsc.br/>>

obrigações acadêmicas e administrativas.

Cada currículo Lattes deve ser acessado individualmente para que os dados necessários sejam coletados. Além disso, cada acesso a um currículo na própria plataforma Lattes deve ser obrigatoriamente precedido pela inserção de um código apresentado em uma imagem, o que atrasa ainda mais o processo. Se os currículos sempre forem acessados através da plataforma, este é um obstáculo que será encontrado cada vez que uma informação deva ser levantada ou até verificada. Por outro lado, mesmo que extraídos e salvos, o acesso a cada documento de forma individual não é uma alternativa célere ou prática. O isolamento dos dados dificulta uma comparação direta entre professores além de não fornecer uma visão geral do contexto do programa. Não há nenhum tipo de sistema que os reúna e disponibilize estas informações de forma unificada ao usuário.

Apesar de não auxiliar as demais alternativas de controle de qualidade, a própria plataforma Sucupira oferece opção de importação de algumas informações dos currículos Lattes, o que já diminui o volume de trabalho necessário para seu preenchimento. No entanto, as informações possíveis de serem importadas são as produções intelectuais, como artigos, livros e trabalhos em anais. Estas não constituem a completude dos dados necessários, ainda que as informações remanescentes estejam disponíveis nos currículos.

Alguns outros sistemas desenvolvidos com o objetivo de auxiliar a pós-graduação foram analisados no decorrer do desenvolvimento deste trabalho. Dentre eles, muitos têm como foco o corpo discente. Sendo assim, os tipos de relatórios gerados por eles, como histórico de disciplinas cursadas, não fornecem informações que contribuem no aprimoramento dos programas. Estes sistemas normalmente não apresentam qualquer integração com a plataforma Lattes. Já os sistemas que são mais orientados à pesquisa são dificilmente adaptados para utilização em outras instituições de ensino.

A plataforma Lattes permite que todos os currículos presentes nela sejam exportados em formato XML. Dados presentes em documentos deste formato podem ser coletados através de uma linguagem de consulta denominada *XQuery*. Esta linguagem, portanto, se torna uma ferramenta essencial na obtenção automatizada das informações desejadas ao permitir que elas sejam coletadas de arquivos extraídos diretamente da plataforma Lattes. Sendo assim, utilizá-la implica em uma aceleração do processo de coleta, atualmente feito de forma manual.

O objetivo deste trabalho é desenvolver um sistema que automa-

tize a geração de relatórios e quantitativos referentes a um programa de pós-graduação através de consultas *XQuery* sobre dados de currículos Lattes no formato XML. Os objetivos específicos deste trabalho são:

- Desenvolver consultas cujos resultados contribuam no controle de qualidade externo e interno dos programas;
- Permitir que as consultas desenvolvidas sejam filtradas por período de tempo;
- Facilitar a aplicação do sistema em diferentes instituições de ensino.

Para tal, propõe-se o desenvolvimento do Ceifador, um sistema *desktop* que coleta os dados referentes a currículos em formato XML previamente adicionados a seu banco de dados, 'ceifando' informações redundantes ou irrelevantes. O Ceifador envia à sua base de dados uma consulta pré-definida escolhida pelo usuário e mostra a ele os resultados obtidos, permitindo ainda que sejam filtrados de acordo com o período de tempo desejado. Estes resultados podem, então, ser salvos ou imediatamente avaliados e utilizados da forma desejada pelo usuário do sistema.

Inicialmente, sistemas similares foram analisados a fim de identificar o funcionamento dos sistemas de apoio à pós-graduação já existentes. A partir disso, características importantes foram levantadas e utilizadas como diretrizes para o desenvolvimento do Ceifador. Dados relevantes a serem obtidos foram concluídos a partir de reuniões com a coordenadoria de pós-graduação e as consultas correspondentes foram posteriormente desenvolvidas. Por fim, após uma análise dos dados obtidos e da melhor forma de apresentação, a interface foi implementada.

O desenvolvimento deste trabalho está dividido em quatro capítulos. A *fundamentação teórica* explica conceitos básicos essenciais ao seu entendimento. O capítulo *trabalhos relacionados* descreve sistemas similares, analisando-os segundo as características levantadas como diferenciais do Ceifador. O *Ceifador* descreve o desenvolvimento deste trabalho, explicando suas aplicações, as tecnologias utilizadas, a forma de implementação da interface e das consultas e desafios encontrados durante seu desenvolvimento. Por fim, o capítulo de *conclusões* analisa se os objetivos foram alcançados e o problema foi resolvido, além de apontar trabalhos a serem realizados no futuro.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns conceitos básicos cuja compreensão é essencial ao entendimento deste trabalho como um todo.

### 2.1 CURRÍCULO LATTES

O Currículo Lattes é um modelo padrão adotado no Brasil para o registro de currículos de pesquisadores brasileiros (CNPQ, 2010). Foi criado sob encomenda do CNPq, o Conselho Nacional de Desenvolvimento Científico e Tecnológico, por grupos universitários em colaboração com profissionais especializados. Os objetivos eram facilitar avaliações curriculares, formar uma base de dados que possibilitasse a seleção de consultores e especialistas e gerar estatísticas sobre a distribuição da pesquisa científica no Brasil.

Alguns formulários eletrônicos para registro de currículos de pesquisadores brasileiros foram desenvolvidos pelo CNPq entre os anos 80 e 90. Diversos fatores distintos, como o funcionamento sobre um sistema operacional específico ou o estado ainda inicial do uso da Internet no Brasil, impediram que sua utilização fosse disseminada. Em 1999, com a Internet mais avançada e a popularização do sistema operacional *Windows*, o Currículo Lattes foi lançado, visando integrar todas as demais versões de currículos já existentes.

Desde seu lançamento, o Currículo Lattes vem aumentando sua abrangência. Atualmente, além de ser utilizado pelas principais universidades, institutos, centros de pesquisa e fundações de amparo à pesquisa brasileiras, também já foi implantado, em uma versão em língua espanhola, por países como Colômbia, Equador, Chile, Peru e Argentina.

Os documentos presentes na Plataforma Lattes estão no formato XML e respeitam um DTD (documento de definição de tipo) específico. Este DTD explicita quais elementos e atributos podem ou não existir, assim como quais são ou não obrigatórios. A Figura 1 representa um fragmento de Currículo Lattes no formato XML.

```

- <CURRICULO-VITAE SISTEMA-ORIGEM-XML="LATTES_OFFLINE" NUMERO-IDENTIFICADOR="0378897709136226"
  DATA-ATUALIZACAO="27102016" HORA-ATUALIZACAO="130551">
+ <DADOS-GERAIS NOME-COMPLETO="Carina Friedrich Dorneles" NOME-EM-CITACOES-
  BIBLIOGRAFICAS="DORNELES, C. F.; Dorneles, Carina F.; Dorneles, Carina Friedrich" NACIONALIDADE="B" PAIS-DE-
  NASCIMENTO="Brasil" UF-NASCIMENTO="RS" CIDADE-NASCIMENTO="Cerro Largo" PERMISSAO-
  DE-DIVULGACAO="NAO" DATA-FALECIMENTO="" SIGLA-PAIS-NACIONALIDADE="BRA" PAIS-DE-
  NACIONALIDADE="Brasil"></DADOS-GERAIS>
+ <PRODUCAO-BIBLIOGRAFICA></PRODUCAO-BIBLIOGRAFICA>
+ <PRODUCAO-TECNICA></PRODUCAO-TECNICA>
- <OUTRA-PRODUCAO>
- <ORIENTACOES-CONCLUIDAS>
  - <ORIENTACOES-CONCLUIDAS-PARA-MESTRADO SEQUENCIA-PRODUCAO="279">
    <DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO NATUREZA="Dissertação de mestrado"
      TIPO="ACADEMICO" TITULO="Recuperação de Informação Temporal" ANO="2009" PAIS="Brasil"
      IDIOMA="Português" HOME-PAGE="" FLAG-RELEVANCIA="NAO" DOI="" TITULO-INGLES=""/>
    <DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO TIPO-DE-
      ORIENTACAO="CO ORIENTADOR" NOME-DO-ORIENTADO="Edimar Manica" CODIGO-
      INSTITUICAO="019200000005" NOME-DA-INSTITUICAO="Universidade Federal do Rio Grande do Sul" CODIGO-
      ORGAO="" NOME-ORGAO="" CODIGO-CURSO="42000041" NOME-DO-CURSO="Computação"
      FLAG-BOLSA="NAO" CODIGO-AGENCIA-FINANCIADORA="" NOME-DA-AGENCIA="" NUMERO-
      DE-PAGINAS="" NUMERO-ID-ORIENTADO="" NOME-DO-CURSO-INGLES="Computer Science"/>
    </ORIENTACOES-CONCLUIDAS-PARA-MESTRADO>
    + <ORIENTACOES-CONCLUIDAS-PARA-MESTRADO SEQUENCIA-PRODUCAO="312"></ORIENTACOES-
      CONCLUIDAS-PARA-MESTRADO>
    (...)
  </OUTRA-PRODUCAO>
+ <DADOS-COMPLEMENTARES></DADOS-COMPLEMENTARES>
</CURRICULO-VITAE>

```

Figura 1 – Fragmento de Currículo Lattes no formato XML

## 2.2 XML

A *eXtensible Markup Language*, ou XML, é uma linguagem de marcação desenvolvida para ser simultaneamente compreensível por seres humanos e por máquinas. Por conta própria, esta linguagem não faz nada, sendo utilizada apenas para armazenar e transferir dados. É necessário que outro *software* opere sobre um documento XML para que as informações armazenadas nele sejam enviadas, recebidas, armazenadas ou exibidas da forma desejada. A diferença da XML para outras linguagens de marcação é que, como o próprio nome já diz, ela é extensível. Isto significa que as aplicações que operam sobre um determinado documento continuam a funcionar mesmo que dados sejam adicionados ou removidos. A adição de novos dados não interfere no acesso a dados já existentes, e a busca por dados removidos retorna nodos vazios. A XML passou a ser recomendada pelo W3C, o *World Wide Web Consortium*, a partir de 10 de fevereiro de 1998 (FAWCETT; AYERS; QUIN, 2012).

Um documento XML possui uma estrutura similar a uma árvore, como pode ser visto na Figura 1. Uma *tag* inicial representa a raiz do documento. Dentro desta, novas *tags* podem ser definidas, representando os nodos filhos, que podem ter novos filhos, e assim sucessivamente. Vale ressaltar que toda *tag* aberta deve ser fechada, e elas devem ser fechadas segundo um ordenamento LIFO (*Last In, First Out*). Além disso, XML é uma linguagem *case-sensitive*, portanto a grafia da abertura e do fechamento das *tags* deve ser a mesma. XML



não usa *tags* pré-definidas. Cada autor pode nomeá-las conforme a conveniência ao seu documento.

Um documento XML contém elementos. Um elemento inclui todo conteúdo existente entre a abertura de uma *tag* e seu fechamento, inclusive. Um elemento pode conter texto, atributos, outros elementos ou uma mistura destes. Na comparação com a estrutura de árvore, cada elemento é similar a um nodo. Também é possível que um elemento esteja vazio. Atributos, por sua vez, são utilizados para armazenar dados referentes aos elementos. Estes podem ser representados também através de outro elemento contido no primeiro. Não há regra em XML que defina o uso de elementos ou atributos, mas há algumas diferenças. Enquanto elementos podem conter múltiplos valores e estruturas de árvore, os atributos não podem.

É possível, também, enrijecer a estrutura de um documento XML, definindo-a através de um DTD ou de um XML *Schema*. Estes definem quais elementos devem estar no documento, quais seus sub-elementos e até seus tipos. Vale ressaltar que todos os dados presentes num documento XML são armazenados em formato de texto. Um DTD ou um *Schema* modifica apenas a forma como eles são interpretados. Dados presentes num documento XML sem esta declaração são considerados sem tipo (MICROSOFT, 2016).

### 2.3 XQUERY

*XQuery* é uma linguagem de consulta a dados XML. É similar à SQL, a *Structured Query Language*, para bancos de dados relacionais. É construída sobre *XPath*, uma sintaxe utilizada para navegar através dos elementos e atributos de um documento XML. A versão 1.0 passou a ser recomendada pelo W3C em 23 de janeiro de 2007, e a versão 3.0 em 8 de abril de 2014 (WALMSLEY, 2016).

*XQuery* reconhece os comandos de *XPath* e possui funções próprias, mas suas consultas são baseadas principalmente em expressões FLWOR. FLWOR é um acrônimo em que cada letra representa um dos cinco comandos principais da linguagem, sendo eles: *For*, *Let*, *Where*, *Order by* e *Return*. O nome de variáveis utilizadas deve ser iniciado pelo caractere '\$'. A Figura 2 representa um exemplo de consulta sobre o documento XML apresentado na Figura 1.

O comando *for* seleciona uma sequência de nodos, sendo que um nodo pode ser um elemento, atributo, texto, ou até a raiz de um documento. Ele vincula uma variável para cada elemento retornado

```

1 | for $o in doc("curriculo.xml")//ORIENTACOES-CONCLUIDAS-PARA-MESTRADO
2 | let $n := $o/DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@NOME-DO-ORIENTADO,
3 | $a := $o/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@ANO
4 | where $a = ("2017")
5 | order by $n
6 | return <p>{data($n)}</p>

```

Figura 2 – Consulta *XQuery* sobre Currículo Lattes em formato XML

pela sequência de nodos informada, resultando em iteração. Assim, pode-se colocar uma sequência numérica neste comando e controlar o número de iterações.

O comando *let* vincula uma sequência a uma variável. Esta sequência pode conter um, vários, ou até nenhum nodo. Utilizando-o, evita-se a repetição de instruções de acesso aos nodos que a variável passou a representar. O comando *let* não resulta em iteração.

O comando *where* filtra os nodos de acordo com um ou mais critérios.

*Order by* ordena os resultados obtidos. É possível ordená-los de acordo com mais de um critério.

*Return*, por fim, indica o que é enviado ao usuário.

A consulta representada na Figura 2 demonstra o uso de todos os comandos. Ela retorna, em ordem alfabética, o nome dos alunos de mestrado concluído no ano de 2017 que foram orientados pelo professor cujo currículo está no arquivo referenciado.

### 3 TRABALHOS RELACIONADOS

Esta seção do trabalho é dedicada à descrição de diversos trabalhos relacionados a este e a uma posterior análise comparativa. Os sistemas aqui apresentados foram analisados sob cinco aspectos distintos, sendo eles: utilização de código aberto, capacidade de gerar relatórios, capacidade de filtrar os dados obtidos, integração com a plataforma Lattes e extensibilidade.

Utilização de código aberto implica no desenvolvimento de um programa permitindo que qualquer pessoa o utilize, estude, redistribua ou melhore (GNU, 2017). Isto possibilita a qualquer usuário adaptá-lo conforme suas necessidades específicas, além de poder realizar a própria manutenção sem depender do auxílio técnico do distribuidor, que pode ou não ser oferecido conforme conveniência. É muito comum, inclusive, que a própria comunidade auxilie na resolução de problemas e no aprimoramento de *softwares* de código aberto.

A análise dos trabalhos em relação à capacidade de gerar relatórios vai além do que o nome implica. Muitos sistemas oferecem esta funcionalidade, mas os relatórios gerados não são, por muitas vezes, úteis na coleta de informações relevantes às avaliações dos programas. Muitos deles fornecem relatórios de listagens de turmas, alunos ou professores. Sendo assim, aqui foi analisada a capacidade de gerar relatórios que possam ser diretamente reutilizados em avaliações de qualidade, como listas de publicações e prêmios, por exemplo.

Capacidade de filtrar os dados obtidos consiste em fornecer ao usuário opções que limitem os dados retornados. Como algumas das aplicações propostas pelo Ceifador incluem auxílio ao levantamento de dados enviados para o acompanhamento anual e a avaliação quadrienal da CAPES, é interessante que os dados retornados já pertençam ao intervalo de tempo desejado. Caso contrário, o usuário ainda deve filtrar manualmente os dados retornados.

Utilização direta do conteúdo da plataforma Lattes implica em não precisar realimentar o banco de dados relacional de um sistema com os dados contidos nos currículos. Esta realimentação implica na total coleta dos dados existentes neles e na posterior inserção no novo banco de dados. Estes passos são desnecessários quando os dados podem ser retirados diretamente dos documentos XML correspondentes ou resgatados de forma automatizada da plataforma Lattes.

Por fim, extensibilidade é a capacidade de facilmente adaptar o programa para funcionar em instituições distintas. A proposta deste

trabalho é desenvolver um sistema que possa ser utilizado por qualquer programa de pós-graduação, de forma que a facilidade de configuração e preparação do ambiente é essencial.

### 3.1 SAPOS

O SAPos - Sistema de Apoio à Pós - é uma aplicação *web* totalmente livre e aberta para gestão de sistemas de pós-graduação, tendo sido desenvolvido para atender às necessidades do Programa de Pós-Graduação em Informática da Universidade Federal do Paraná. Ele é capaz de gerar diversos relatórios, mas estes dizem respeito majoritariamente aos alunos ou à relação dos professores com eles, como orientações ou participações em bancas. A evolução do sistema é responsabilidade dos próprios alunos de Ciências da Computação da Universidade, mas contribuições feitas pelo público são aceitas, tanto de forma pontual como periódica. As informações sobre este sistema foram retiradas da página do sistema e em trocas de *e-mails* com o desenvolvedor (FABRO, 2011).

Em relação às características observadas:

- Seu código está disponível para o público em uma plataforma GitLab<sup>1</sup> da própria universidade;
- O sistema emite alguns relatórios que podem ser utilizados na coleta de informações para o controle de qualidade, como as orientações concluídas e em andamento dos professores;
- As descrições dos relatórios gerados não informam capacidade de aplicação de filtros;
- A utilização de dados da plataforma Lattes, apesar de planejada, ainda não foi implementada, o que implica na necessidade de atualização dos dois sistemas de forma independente;
- O sistema é parametrizável e internacionalizável, o que permite que seja utilizado em instituições distintas.

Existe outro sistema para apoio à gestão de programas de pós-graduação denominado SAPOS desenvolvido pela Universidade Federal Fluminense. Para este trabalho optou-se por analisar apenas o primeiro devido à quantidade de informações disponíveis.

---

<sup>1</sup><https://gitlab.c3sl.ufpr.br/c3sl/sapos>

### 3.2 COBALTO

O Cobalto é o Sistema Integrado de Gestão Acadêmico-Administrativa da Universidade Federal de Pelotas. Foi desenvolvido pela equipe de servidores de Tecnologia da Informação da própria universidade, sendo aprimorado para englobar cada vez mais áreas da comunidade acadêmica. Seu foco é o auxílio à administração de ensino na forma de matrículas e atribuição de notas. Os relatórios gerados também têm esta característica, incluindo relatórios de alunos que são beneficiados por auxílio deslocamento, por exemplo. O sistema oferece diversas funcionalidades a alunos, professores e também servidores administrativos. As informações sobre este sistema foram retiradas de sua *wiki* (UFPEL, 2010).

Observou-se que:

- Ele é baseado na política brasileira de software livre, tendo seu código atualmente disponível na plataforma GitHub<sup>2</sup>;
- Apesar de oferecer a opção de gerar relatórios, estes não podem ser utilizados no controle de qualidade, pois têm caráter principalmente administrativo, como alunos ingressantes ou com débito e turmas ofertadas;
- O Cobalto permite que seus usuários filtrem os relatórios gerados de acordo com um período específico;
- O sistema oferece informações referentes a projetos de pesquisa, mas todo novo projeto deve ser cadastrado manualmente. Não existe ainda uma integração com a Plataforma Lattes, de forma que cada um dos dois deve ser atualizado individualmente;
- O sistema disponibiliza módulos utilizáveis pela plenitude da comunidade acadêmica da Universidade Federal de Pelotas, mas ainda é exclusivo desta. Apesar de ter seu código livremente acessível, outras instituições ainda não incorporaram o sistema.

### 3.3 SIGPOS

O SigPos - Sistema de Gestão da Pós-Graduação - é o sistema utilizado na Universidade Federal do Mato Grosso do Sul para auxílio

---

<sup>2</sup><https://github.com/cobalto-php/cobalto-php>

administrativo a seus cursos de especialização, mestrado e doutorado. Os relatórios gerados são focados nos alunos, havendo poucas opções para dados de docentes, como participações em bancas. Seus desenvolvimento e manutenção são responsabilidade do Núcleo de Tecnologia da Informação da instituição. As informações sobre este sistema foram retiradas de seu manual (UFMS, 2013).

Sobre o SigPos:

- Seu código não é disponível para a comunidade;
- O sistema oferece opções de relatórios focados nos alunos, como atestado de matrícula e histórico escolar, mas não oferece informações que possam ser utilizadas no controle de qualidade;
- O sistema permite que filtros sejam aplicados aos relatórios gerados;
- O sistema oferece um controle de docentes. No cadastro de novos professores, uma das informações solicitadas (mas não obrigatória) é o *link* para seu currículo Lattes. Esta é a única referência à plataforma, não havendo integração direta entre os dados;
- Não há qualquer indicação de utilização do SigPos em alguma outra instituição.

### 3.4 SIPEX

O SIPEX - Sistema de Informação de Pesquisa e Extensão - é um sistema utilizado na Universidade Estadual de Campinas cujo objetivo é facilitar a gestão das produções científicas geradas por docentes, discentes e pesquisadores. Os relatórios gerados pelo SIPEX apresentam um foco maior nos docentes, havendo relatórios de artigos publicados, por exemplo. A UNICAMP conta com um grupo gestor responsável por definir os objetivos, escopos, conceitos e diretrizes gerais do SIPEX. As informações sobre este sistema foram retiradas da seção de perguntas frequentes de seu *site* (UNICAMP, 2017).

Sobre as características observadas:

- Seu código não é disponibilizado para a comunidade;
- O sistema gera relatórios referentes às produções artísticas, científicas e tecnológicas dos usuários cadastrados, o que pode ser usado no controle de qualidade;

- O SIPEX permite que os dados sejam filtrados informando períodos de tempo desejados;
- Todos os dados presentes no SIPEX podem ser diretamente importados ou exportados para a plataforma Lattes;
- O sistema não foi adotado por outras instituições.

Existe outro sistema denominado SIPEX utilizado na Universidade Regional de Blumenau, mas eles são independentes. Como anteriormente, para este trabalho optou-se por analisar apenas o primeiro pela quantidade de informações disponíveis.

### 3.5 SIGA

O SIGA é o Sistema de Gestão da Pós-Graduação da UFPR, oferecendo informações sobre atividades de pesquisa e bolsas de ensino. Foi planejado como uma proposta de unificação dos *softwares* utilizados para a gestão acadêmica, facilitando assim o trabalho de envio de informações para a CAPES. Sendo assim, os relatórios gerados por ele são específicos para auxiliar o levantamento dos dados necessários às avaliações oficiais. Ele foi inicialmente desenvolvido como um Trabalho de Conclusão de Curso, mas atualmente apresenta uma equipe específica para sua manutenção e desenvolvimento. As informações sobre este sistema foram retiradas da versão escrita do trabalho desenvolvido (SCOTTINI et al., 2014).

Sobre o SIGA:

- Apesar de ter sido desenvolvido como o TCC de uma universidade pública, e o código estar inicialmente disponível, o programa continuou sendo aprimorado e o código atualizado não pode ser encontrado (UFPR, 2016);
- Permite que os usuários extraiam diversos relatórios que podem ser utilizados na avaliação oficial da CAPES e que também podem ser utilizados no controle de qualidade interno, como listagem de projetos de pesquisa e produções intelectuais de um determinado usuário;
- Os relatórios gerados não apresentam nenhuma opção de filtro;
- O SIGA importa informações como grupos, equipes e linhas de pesquisa da plataforma Lattes;

- Como o próprio título implica, o sistema foi desenvolvido para ser utilizado especificamente pela UFPR. Nenhuma outra instituição o adotou.

### 3.6 DISCUSSÃO E ANÁLISE COMPARATIVA

A Tabela 1 mostra uma comparação dos sistemas discutidos e do trabalho aqui apresentado. A análise foi feita segundo as características descritas anteriormente.

	SAPos	Cobalto	SigPos	SIPEX	SIGA	Ceifador
Código aberto	X	X				X
Relatórios	X			X	X	X
Filtros		X	X	X		X
Lattes				X	X	X
Extensibilidade	X					X

Tabela 1 – Comparação entre os trabalhos relacionados

O Ceifador tem seu código disponível. Assim, cada usuário ou instituição que escolha utilizá-lo pode modificar, adaptar ou expandir suas funcionalidades conforme o desejado. Desta forma, usuários com demandas mais específicas ainda podem aproveitá-lo.

O objetivo principal do sistema é auxiliar os programas de pós-graduação no controle de qualidade, nas avaliações e na divulgação de quantitativos especificamente através dos relatórios gerados. A definição destes foi feita a partir do levantamento de requisitos realizado, garantindo assim que todo relatório gerado pelo Ceifador tem aplicação prática no controle de qualidade do programa. Muitos dos dados obtidos podem ser diretamente utilizados no preenchimento da Plataforma Sucupira.

A opção de utilização de filtros temporais expande as aplicações do sistema. Com ela, é possível que sejam obtidos dados do último triênio, quadriênio, do último ano, ou até fazer uma comparação entre as produções acadêmicas dos professores antes e depois de ingressarem no corpo docente do programa. Os filtros temporais são essenciais no ceifar de dados que não correspondem aos desejos imediatos do usuário, não havendo necessidade de realizar uma recoleta de informações sobre um relatório gerado.

A utilização dos dados presentes no Lattes é uma característica



essencial ao Ceifador. Apesar de não resgatar os dados da plataforma de forma automática, ele não necessita que toda informação presente em cada currículo seja realimentada a um banco de dados relacional. Basta que os currículos extraídos da plataforma Lattes em formato XML sejam inseridos nele para que as informações possam ser extraídas. Isto garante também sua extensibilidade, bastando que os currículos referentes a professores de um programa distinto sejam inseridos para que as novas informações sejam extraídas. O sistema pode ainda ser configurado para se comunicar com mais de uma base de dados. Assim, basta que um banco de dados distinto seja utilizado para que as informações referentes a outro programa sejam resgatadas, não havendo necessidade de modificar completamente a que está sendo utilizada.



## 4 O CEIFADOR

Este capítulo visa discutir os avanços realizados em direção ao cumprimento da proposta inicial e dos objetivos deste trabalho. O código desenvolvido está disponível na plataforma GitHub<sup>1</sup>.

### 4.1 VISÃO GERAL

O Ceifador tem por objetivo coletar informações através de consultas aos dados presentes nos currículos inseridos nele, 'ceifando' aquelas consideradas redundantes ou irrelevantes. Toda interação entre o usuário e a base de dados é feita através de uma interface gráfica. O sistema fornece ao usuário opções de consultas cujos resultados podem ser úteis para fins que vão desde a divulgação de quantitativos do programa de pós-graduação até o auxílio no preenchimento da Plataforma Sucupira de avaliação da CAPES. A Figura 3 representa uma visão geral do funcionamento do Ceifador.

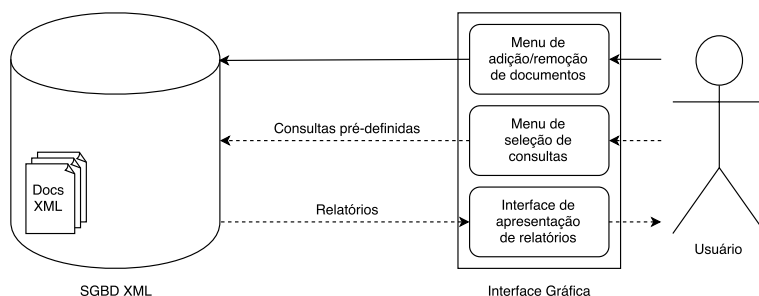


Figura 3 – Visão geral do funcionamento do sistema

A interface permite que o usuário selecione o relatório desejado. Cada opção de relatório oferecida pelo sistema tem um botão correspondente na interface. Este, quando selecionado, faz com que uma consulta pré-definida seja enviada ao servidor e as informações desejadas sejam posteriormente apresentadas pela interface. Os resultados não são apenas exibidos, mas também podem ser exportados. Desta forma, o usuário pode armazená-los para posterior análise sem que o sistema seja acessado novamente. A interface permite também a modificação

<sup>1</sup><https://github.com/bsvalverde/Ceifador>

do conteúdo presente no banco de dados utilizado.

A base de dados do Ceifador é constituída por documentos XML referentes a Currículos Lattes. Caso o usuário deseje verificar os documentos que já a compõem, estes são exibidos por ordem alfabética acompanhados da data da última atualização. Vale ressaltar que esta data é referente à última atualização do documento, e não da base de dados. Para excluir um documento, basta selecioná-lo e clicar no botão correspondente. Para adicionar novos documentos, é necessário encontrar o arquivo corresponde no computador utilizado. Para atualizar um currículo cujos dados estão ultrapassados, é necessário remover o documento anterior e adicionar um novo.

É permitido ao usuário estabelecer um filtro temporal que é aplicado às consultas relevantes. Quando utilizado, este delimita o ano inicial e final aos quais os dados devem pertencer. Caso ignorado, todas as informações são retornadas.

#### **4.1.1 Aplicações**

As principais aplicações do Ceifador se dão no âmbito do controle de qualidade dos programas de pós-graduação, tanto oficiais quanto extraoficiais. O meio de controle de qualidade oficial é a avaliação quadri-  
enal realizada pela CAPES, que pode impedir os programas de abrirem novas turmas caso não alcancem nota igual ou superior a 3, em uma escala de 1 a 7. Já como meios extraoficiais pode-se considerar medidas internas que os programas adotem a fim de manter elevado o nível de ensino e ampliar seus recursos, como um planejamento estratégico, por exemplo.

O controle de qualidade oficial da CAPES é feito através do preenchimento da Plataforma Sucupira. Alguns dos dados solicitados são a diversidade das áreas de estudo do corpo docente, assim como suas produções técnicas e bibliográficas. Outro fator importante é a distribuição destas características. O ideal é que todas as áreas de estudos tenham um número similar de professores, e também que as publicações do programa estejam bem distribuídas entre eles. Poucas áreas de estudos com muitos professores e algumas com poucos não são uma característica positiva, assim como poucos professores com muitas publicações e outros com poucas. Todas estas informações, juntamente com quantitativos, estão disponíveis nos relatórios gerados pelo Ceifador, bastando assim que os dados presentes neles sejam transcritos para a plataforma Sucupira.

Como forma de controle de qualidade extraoficial e interna a cada programa de pós-graduação pode-se destacar o planejamento estratégico para a definição de objetivos e meios de alcançá-los. Nestes são observadas as publicações, orientações e prêmios recebidos pelos professores, assim como outras características que possam auxiliar na definição de meios para atrair mais estudantes, parceiros ou financiadores. Através da análise da produção docente, a coordenadoria consegue definir meios para a expansão do programa.

A opção de aplicação de filtros temporais expande as aplicações do sistema. Com ela, é possível analisar o progresso do programa, não sendo oferecida apenas uma visão geral do mesmo. No caso específico de um planejamento estratégico é possível não apenas definir objetivos e meios de alcançá-los, mas também analisar o progresso e as metas alcançadas desde a última avaliação interna. Caso uma avaliação determine que um professor deve aumentar o número de artigos publicados, por exemplo, uma avaliação subsequente pode gerar uma consulta que retorne apenas os artigos publicados por ele no último ano. Assim, não é necessário que o usuário faça a diferenciação de artigos antigos e novos por conta própria, além de um total numérico ser oferecido pelo próprio sistema. Outra facilidade oferecida pelo sistema é a visualização da data de última atualização dos currículos utilizados, permitindo que a coordenadoria tenha garantia de gerar seus relatórios a partir dos dados mais atuais.

Como o Ceifador conta os resultados obtidos e fornece estes valores ao usuário, ele pode ter uma aplicação secundária no auxílio à publicidade do programa. Estes quantitativos podem ser utilizados em divulgações de produtividade, o que pode ser visto como um atrativo para potenciais alunos ou parceiros e financiadores de projetos de pesquisa. Atrair financiadores é muito importante, pois pode auxiliar na internacionalização do programa e aumentar os recursos disponíveis. Isto implica em ampliar os meios de obter progresso e resultados em pesquisa, que é a principal finalidade de um programa de pós-graduação.

## 4.2 TECNOLOGIAS UTILIZADAS

Esta seção visa descrever o SGBD e a linguagem de programação utilizada no desenvolvimento da interface, assim como a API escolhida para conectá-los.

### 4.2.1 *BaseX*

*BaseX* é um sistema gerenciador de bancos de dados XML escalável e de alta performance, além de ser um processador de *XPath/XQuery*. Ele oferece suporte às mais recentes recomendações W3C e *Full Text* (BASEX, 2017). O foco deste SGBD é permitir que o usuário armazene, consulte e visualize grandes volumes de documentos XML e JSON.

*BaseX* pode ser executado de quatro formas distintas. A primeira é uma execução autônoma, que permite que consultas sejam feitas utilizando a linha de comando. Outra forma possível é através de interface gráfica, em que os comandos de gerência do banco, como criação de uma nova base ou adição e remoção de documentos, são substituídos por botões. Esta interface permite também que os dados sejam explorados de forma interativa e as expressões *XQuery* sejam avaliadas em tempo real. A terceira forma é em um esquema cliente/servidor. O modo servidor foi desenvolvido para permitir que o *BaseX* seja utilizado por mais de um usuário simultaneamente. Como servidor, *BaseX* gerencia a concorrência entre operações de leitura e escrita e informações de usuário, além de manter *logs* de todas as transações realizadas. Como cliente, pode enviar comandos e consultas para o servidor através da linha de comando, de forma similar à execução autônoma. Por fim, *BaseX* pode também ser executado como um servidor HTTP, sendo assim utilizado como aplicação *web*.

*BaseX* é independente de plataforma e distribuído sob a licença BSD (*Berkeley Software Distribution*).

### 4.2.2 *Java*

*Java* é uma linguagem de programação orientada a objetos com seu próprio ambiente de execução. Isto significa que programas escritos em *Java* são independentes de plataforma, sendo capazes de rodar em qualquer sistema operacional com qualquer tipo de processador desde que o interpretador necessário esteja disponível. Outra vantagem é que um programa *Java* que funciona em uma plataforma não precisa ser recompilado para funcionar em outra. Apesar disso, a JVM (*Java Virtual Machine*), que executa os códigos *Java*, é escrita em outras linguagens de programação, e portanto é dependente de plataforma. Ainda assim, a principal característica associada a esta linguagem é portabilidade (GOSLING; MCGILTON, 1995).

Os requisitos do desenvolvimento da linguagem *Java* foram determinados pela natureza do ambiente computacional no qual novos *softwares* devem sobreviver. Para que sejam compatíveis com um contexto de comércio e distribuição eletrônicos, a tecnologia *Java* permite o desenvolvimento de aplicações seguras e de alta performance e robustez que funcionam em múltiplas plataformas e em redes distribuídas e heterogêneas. *Java* foi desenvolvida para ser facilmente programada, para que seja facilmente aprendida por programadores que já utilizem outras linguagens, para se enquadrar em aplicações cliente-servidor distribuídas, para oferecer alta performance em aplicações que precisam executar mais de uma atividade simultaneamente e para oferecer máxima capacidade de portabilidade e dinamicidade.

Diversos tipos de aplicação podem ser desenvolvidos com esta linguagem. *Java* é utilizada para desenvolver tanto o lado servidor de aplicações *web* quanto aplicações *desktop* independentes. Também é utilizada para desenvolver aplicações distribuídas, como aplicações de bancos e no desenvolvimento de aplicativos para dispositivos móveis, sendo uma das linguagens utilizadas em aplicativos para *Google Android*, por exemplo.

A linguagem *Java* foi escolhida para este trabalho justamente por sua diversidade. Utilizando-a, este programa pode tanto ser utilizado como um programa *desktop* independente quanto inserido como um *applet* em uma página HTML, ampliando assim seu alcance.

### 4.2.3 XQJ

XQJ é uma API *XQuery* para *Java*. Ela funciona como uma interface padrão para que as aplicações se comuniquem com bases de dados XML, enviando a elas instruções de comando ou consultas *XQuery* (XQJ.NET, 2014).

A API permite que os resultados das consultas enviadas à base de dados sejam processados em *Java*, e, também, que variáveis *XQuery* sejam ligadas a variáveis do programa. As transações realizadas através do XQJ contém as propriedades ACID - Atomicidade, Consistência, Isolamento e Durabilidade -, e a API permite leitura e escrita de dados em qualquer codificação.

No caso específico do *BaseX*, XQJ faz com que a aplicação *Java* funcione como o cliente no modelo cliente/servidor, permitindo que comandos e consultas sejam armazenados em *strings* e enviados ao banco.

#### 4.2.4 HTML

A *HyperText Markup Language*, ou HTML, é a linguagem de marcação padrão para o desenvolvimento de páginas e aplicações *web*. Navegadores recebem documentos HTML, interpretam seu conteúdo e os transformam em páginas multimídia. Os documentos descrevem a estrutura semântica da página e, originalmente, também sua aparência (W3SCHOOLS, 2010).

Documentos HTML possuem uma estrutura de árvore similar aos documentos XML, em que seus elementos são definidos por *tags* que podem ou não conter outros elementos. Estes são considerados os blocos básicos utilizados em sua construção. Eles podem introduzir novos conteúdos na página, como imagens, ou simplesmente fornecer informações sobre a formatação do texto que contêm.

Documentos HTML apresentam, entretanto, algumas diferenças em relação à estrutura de documentos XML. A principal delas é que documentos HTML utilizam *tags* pré-definidas. Desta forma, os navegadores sabem como interpretar cada uma delas. Outra diferença é que eles não são *case-sensitive*, compreendendo comandos mesmo que a grafia das *tags* inicial e final seja distinta. O W3C recomenda, no entanto, que a grafia das *tags* seja sempre feita em letras minúsculas. Por fim, algumas das *tags* utilizadas permitem que sua ordem de fechamento não seja exatamente inversa a sua ordem de abertura.

Atualmente, páginas *web* não são mais constituídas apenas por código HTML. Juntamente com CSS - *Cascading Style Sheet* - e *JavaScript*, esta linguagem forma uma tríade de tecnologias base para a *World Wide Web*. CSS é uma linguagem de estilo utilizada para definir a apresentação de um documento escrito em linguagem de marcação, e *JavaScript* é a linguagem de programação utilizada no desenvolvimento de aplicações *web*. O W3C recomenda a utilização do CSS ao invés da definição de estilos no próprio documento HTML desde 1997.

Neste trabalho específico, a linguagem HTML é utilizada na formatação dos relatórios gerados. As consultas *XQuery* desenvolvidas já fornecem, como retorno, código HTML parcialmente pronto para ser interpretado. A Figura 2 representa uma consulta com *tags* HTML em seu retorno. Nela, cada nome de aluno é impresso em um parágrafo distinto.



### 4.3 INTERFACE

Nesta seção é descrito o desenvolvimento da interface *Java* do Ceifador. Ela foi planejada para ser intuitiva e pouco complexa, não havendo transição de janelas ou longos caminhos a serem percorridos para que as funções do programa possam ser utilizadas.

Ao ser iniciado, o Ceifador apresenta uma janela com duas abas, como pode ser visto na Figura 4. Uma destas abas permite que o usuário selecione a consulta desejada. A outra, que manipule os dados do banco.



Figura 4 – Visão inicial do Ceifador mostrando a aba de seleção de consultas

As consultas foram separadas em grupos de acordo com os dados obtidos por elas para facilitar sua localização. Estes grupos são: listas simples, dados dos professores, dados dos professores com seus alunos e dados dos professores com outros professores. Cada consulta disponível está brevemente descrita, para que o usuário possa identificá-la, e acom-

panhada por um botão. Ao clicar no botão, a consulta correspondente é enviada à base de dados.

Há também a opção de aplicar um filtro de tempo aos relatórios gerados. No topo da aba há um botão que, quando clicado, abre uma janela que permite que o usuário insira o ano inicial e final desejado, assim como desative qualquer filtro atualmente aplicado. Esta janela pode ser vista na Figura 5. Quando um filtro está em uso, o texto do botão é modificado para que ele seja devidamente mostrado ao usuário.

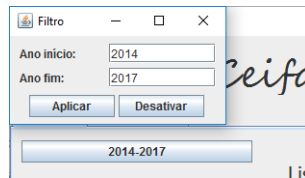


Figura 5 – Aplicação de filtro temporal às consultas

A comunicação entre o código *Java* e a base de dados foi baseada em um *script* disponibilizado pela própria API. O código utilizado é mostrado na Figura 6.

```

120      XQDataSource xqs = new BaseXXQDataSource();
121      xqs.setProperty("serverName", DBConfig.host);
122      xqs.setProperty("port", DBConfig.port);
123
124      XQConnection conn = xqs.getConnection(DBConfig.user, DBConfig.password);
125
126      XQExpression xqe = conn.createExpression();
127      xqe.executeCommand("OPEN " + DBConfig.dbName);
128
129      XQResultSequence rs = xqe.executeQuery(consulta);
130      String result = "";
131      while(rs.next())
132          result += rs.getItemAsString(null);
133
134      xqe.executeCommand("CLOSE");
135
136      conn.close();
137      return result;

```

Figura 6 – Código de comunicação com o *BaseX* utilizando a API XQJ

O primeiro passo é criar um objeto que representa a base de dados (linha 120). Este objeto deve ser configurado com o nome do servidor e a porta que ele ouve (linhas 121 e 122). O servidor do *BaseX* escuta, de acordo com as configurações padrão, a porta 1984. A partir do objeto criado, uma conexão é solicitada, desta vez enviando o usuário e a senha do servidor *BaseX* (linha 124). A partir da conexão, um objeto que executa os comandos e expressões é criado (linha 126).

O *BaseX* é um SGBD que pode armazenar diversos bancos de dados distintos. Sendo assim, quando inicializado, as operações enviadas a ele não são feitas sobre nenhuma base de dados específica. É necessário enviar um comando *OPEN* para que se possa acessar os documentos da base de dados desejada (linha 127). Neste momento é possível enviar uma *string* cujo conteúdo é a consulta *XQuery* para a base de dados (linha 129). O resultado, por sua vez, é representado por outro objeto obtido como retorno do comando anterior. Cada linha do resultado é então transformada em uma *string* e armazenada (linhas 131 e 132). Por fim, a comunicação com o banco é encerrada (linhas 134 e 136). Caso a comunicação com o banco falhe, o usuário é notificado e a consulta não é realizada.

Todos os dados necessários para que a comunicação seja estabelecida (nome do servidor, número da porta, nome de usuário, senha e nome do banco de dados) foram estabelecidos em um arquivo distinto. Isto permite que os dados sejam resgatados e modificados em tempo de execução, além de facilitar sua alteração quando esta for feita diretamente através do código. O fato da conexão ser estabelecida e então fechada cada vez que uma consulta é enviada permite que o banco de dados com o qual se está trabalhando seja alterado sem que o programa tenha que ser encerrado. Todos estes dados podem ser alterados através do botão configurar presente no canto superior esquerdo da tela.

Para mostrar o resultado das consultas ao usuário, uma nova aba é criada. O nome desta é constituído pelo número atribuído à consulta na aba inicial e pelos anos para os quais o resultado foi obtido, caso algum filtro temporal tenha sido utilizado. Os dados obtidos são apresentados sobre um painel de rolamento. O usuário pode exportar o resultado obtido ou fechar a aba através de botões correspondentes. A aba em que o resultado é mostrado pode ser vista na Figura 7.

Como visto na Figura 6, o resultado obtido pelas consultas é um objeto *string*. Esta é inserida como texto em um objeto da classe *Java JLabel*. Esta classe consegue interpretar e mostrar ao usuário textos formatados em HTML. Sendo assim, considerando-se a visibilidade e a estética de tabelas HTML superior às das opções de tabelas disponíveis em *Java*, a *string* obtida da consulta é devidamente configurada para ser exibida conforme a Figura 7. Para que isto seja possível, as *tags* HTML necessárias à implementação das linhas da tabela foram incorporadas no retorno das consultas *XQuery* desenvolvidas. A extensão dos arquivos criados através do botão exportar também é HTML.

Por fim, a aba da base de dados permite que o usuário manipule os documentos presentes nela. Os documentos presentes no banco e

Aldo von Wangenheim	
Prêmio	Ano
Best Paper at SIBGRAPI 2014 - Conference on Graphics, Patterns and Images for the paper "Portable Digital In-line Holography Platform for Sperm Cell Visualization and Quantification"	2014
SET Premio 2014 na categoria Projeto de Interatividade para Televisão - SETEXPO 2014 ? Broadcast and New Media Technology ? Trade Show and Conference	2014
Google RISE Award (concedido à Iniciativa Computação na Escola)	2015
IEEE CBMS 2016 Best Student Paper Award	2016
Total:	4
Antonio Augusto Medeiros Fröhlich	
Prêmio	Ano
Best Paper on Real-time Systems	2015
Best Paper on Operating System	2016
IEEE Senior Member	2017
Total:	3
Carla Merkle Westphall	
Prêmio	Ano

Figura 7 – Visualização do resultado de uma consulta

suas respectivas datas de atualização são coletados através de uma consulta *XQuery* e mostrados ao usuário através de uma tabela, conforme pode ser visto na Figura 8. A consulta que retorna estas informações retorna também o endereço de cada documento dentro da base. Estes são armazenados para que possam ser utilizados posteriormente na função de remoção de arquivos. A formatação da tabela utilizada nesta aba é distinta daquela utilizada para exibir os relatórios gerados. Nesta, linhas podem ser selecionadas. Tendo feito isto, basta clicar no botão de remoção para que o documento correspondente seja removido da base de dados. Esta aba permite também que novos documentos sejam adicionados ao banco. Para tal, basta selecionar o documento correspondente dentre os arquivos do computador e clicar em "Adicionar". O conteúdo da tabela é atualizado após cada operação de adição ou remoção para mostrar corretamente quais documentos estão presentes no banco.

O código utilizado para adicionar e remover documentos da base

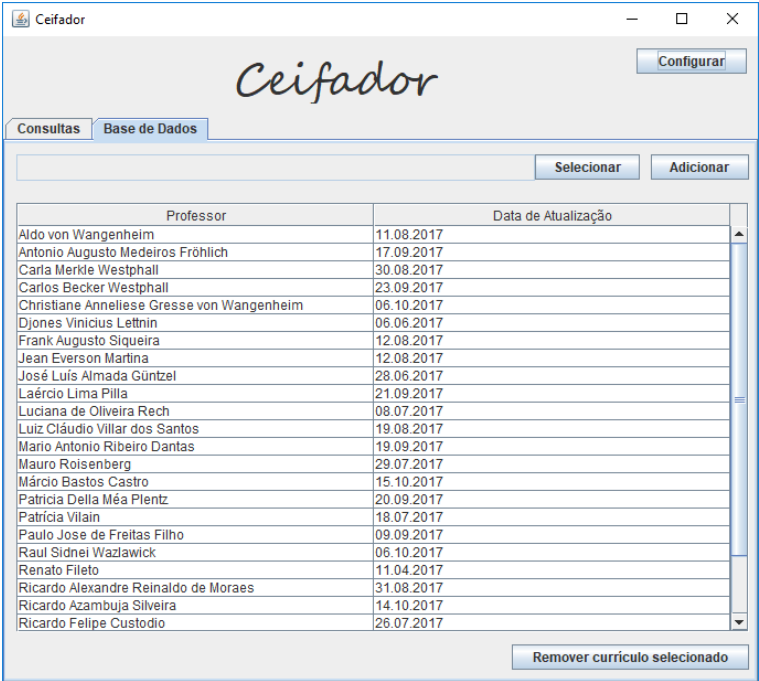


Figura 8 – Aba de manipulação do banco de dados

de dados é similar ao mostrado na Figura 6, substituindo as consultas *XQuery* pelo comando apropriado.

#### 4.4 CONSULTAS

Esta seção visa discorrer sobre as consultas projetadas, tanto enumerando-as quanto explicando o raciocínio por trás de seu desenvolvimento.

##### 4.4.1 Consultas projetadas

A definição das consultas projetadas foi feita através de reuniões com a coordenação do PPGCC UFSC. Nelas, foram levantados dados cuja fácil obtenção auxiliaria no gerenciamento do programa. Sendo assim, as seguintes consultas foram desenvolvidas para a obtenção destes

dados:

1. Lista dos professores;
2. Lista das áreas de atuação dos professores, assim como o número de docentes que atua em cada uma delas;
3. Para cada professor, os artigos já publicados;
4. Para cada professor, os capítulos de livros já publicados;
5. Para cada professor, participantes e financiadores de seus projetos de pesquisa;
6. Para cada professor, os periódicos que revisa e já revisou;
7. Para cada professor, os eventos que organizou;
8. Para cada professor, as premiações recebidas;
9. Para cada professor, os alunos de iniciação científica em andamento ou concluída;
10. Para cada professor, os alunos que foram seus orientandos de TCC e também de mestrado ou doutorado;
11. Para cada professor, os alunos que concluíram o mestrado sob sua orientação;
12. Para cada professor, os alunos de mestrado coautores em artigos;
13. Para cada professor, os alunos que concluíram o doutorado sob sua orientação;
14. Para cada professor, os alunos de doutorado coautores em artigos;
15. Para cada professor, os pesquisadores com quem foi coautor em artigos;
16. Para cada professor, os demais professores do mesmo programa com quem colabora em projetos de pesquisa;
17. Para cada professor, os N professores do mesmo programa com quem mais publicou artigos, assim como o número de artigos publicados em conjunto.

A implementação destas consultas pode ser encontrada no Apêndice A.

As consultas 1, 2 e 17 são as únicas cujos resultados não são influenciados pelo filtro temporal. Além disso, o retorno de todas as consultas inclui um quantitativo referente às informações obtidas. Em consultas cujos resultados podem ser separados como concluídos dentro do período de tempo selecionado ou ainda em andamento, os dados estão devidamente separados, assim como os quantitativos obtidos.

#### 4.4.2 Desenvolvimento das Consultas

Cada consulta desenvolvida deve retornar as linhas de uma tabela contendo os dados relevantes ao relatório selecionado. Estas devem ser coletadas sobre todos os Currículos Lattes pertencentes ao banco e pertencentes ao intervalo de tempo desejado quando assim for possível. Um quantitativo correspondente aos dados coletados também deve ser retornado para cada professor. Os resultados devem ser dispostos de forma organizada para maior legibilidade. Esta seção explica como as consultas foram desenvolvidas a fim de garantir estes requisitos.

A Figura 9 mostra uma das consultas *XQuery* desenvolvidas para este trabalho. Ela retorna, para cada professor do PPGCC UFSC, as premiações recebidas por ele dentro do intervalo de ano definido.

```

1 for $doc in collection()
2 let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
3 $c := (for $p in $doc//PREMIO-TITULO
4   where $p/@ANO-DA-PREMIACAO = ("2014", "2015", "2016", "2017")
5   return $p)
6 order by $name
7 return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
8   <tr><th bgcolor="#C7C7C7">Prêmio</th><th bgcolor="#C7C7C7">Ano</th></tr>,
9   for $p in $c
10  let $pn := $p/@NOME-DO-PREMIO-OU-TITULO,
11  $pa := $p/@ANO-DA-PREMIACAO
12  order by $pa, $pn
13  return <tr><td>{data($pn)}</td><td>{data($pa)}</td></tr>,
14  <tr><td bgcolor="#C7C7C7">Total:</td><td>{count($c)}</td></tr>
15 )

```

Figura 9 – Consulta *XQuery* que retorna, para cada professor, a lista e o número total de premiações recebidas

Inicialmente, deve-se garantir que a consulta é feita sobre toda a base de dados. Para tal, foi utilizada a função *XQuery collection()*, que retorna um nodo para cada recurso disponível no endereço passado. Neste caso, tendo sido utilizada sem parâmetros, esta função retorna a coleção padrão, ou seja, um conjunto de nodos referentes a cada um

dos documentos pertencentes à base de dados. Declarando-se a variável *\$doc* junto do comando *for*, obtém-se uma variável que itera sobre cada currículo Lattes adicionado à base de dados.

Nodos que são reutilizados foram armazenados em variáveis através do comando *let*, como pode ser observado nas linhas 2, 3, 10 e 11. A linha 2, especificamente, armazena o nome do professor ao qual o currículo pertence. Este é novamente utilizado posteriormente na apresentação dos resultados. Cada vez que um nodo ou atributo é descrito, o caminho até ele é percorrido para que o dado possa ser acessado. Ao atribuir seu valor a uma variável, basta que este caminho seja percorrido uma vez. Isto gera uma economia de processamento que pode ser bastante notável caso uma mesma variável seja acessada repetidamente dentro de um comando *for*, por exemplo.

Os dados que são retornados estão armazenados na variável *\$c*, para que possam ser contabilizados. Os nodos desejados, neste caso aqueles referentes às premiações, foram localizados no documento e devidamente filtrados conforme o intervalo de anos definido através de uma consulta aninhada (linhas 3-5). Vale ressaltar que os anos utilizados na imagem servem apenas como exemplo. O sistema modifica estes valores conforme os dados recebidos pelo usuário. Este armazenamento implica também em uma economia de processamento, como mencionado no parágrafo anterior. Caso os prêmios filtrados não fossem armazenados em uma variável, eles deveriam ser acessados duas vezes: uma a fim de coletar os dados para a visualização do usuário e outra para a contagem.

O resultado desejado consiste em uma linha de identificação do professor seguido de uma linha para cada prêmio recebido por ele, estas seguidas pelo total contabilizado. Para tal, o retorno da consulta inicial consiste em mais de um item. Nele, cada item retornado é separado por uma vírgula. Para que o banco saiba que todos eles pertencem ao mesmo comando, os itens são encapsulados por parênteses. Neste caso os retornos são o nome do professor, uma linha de identificação para explicitar quais os dados que são retornados ao usuário, uma subconsulta e uma última linha com o total de dados retornados para aquele professor. Fazendo com que esta subconsulta seja uma nova iteração, é possível adicionar tantas linhas quanto houverem prêmios.

A subconsulta consiste em uma iteração sobre os prêmios já filtrados na consulta mais externa (linha 9). As informações necessárias, neste caso o nome do prêmio e o ano de recebimento, são armazenadas em variáveis (linhas 10 e 11) e devidamente retornadas (linha 14). Os comandos presentes nas linhas 6 e 12 garantem que os resultados estão



devidamente organizados.

Por fim, a função *count()* é utilizada para quantificar os dados obtidos (linha 14).

As tags utilizadas nas linhas 7, 8, 13 e 14 consistem na codificação das linhas da tabela HTML obtida como resultado.

#### 4.5 DESAFIOS

Apesar de cumprir o objetivo de gerar de forma rápida e simples diversos relatórios de auxílio à coordenação da pós-graduação, alguns desafios foram encontrados no desenvolvimento do Ceifador. Estes provêm de informações ausentes em currículos, da forma como estes foram preenchidos e também de limitações do sistema desenvolvido. Esta seção visa discuti-los.

Primeiramente, informações que não tenham sido inseridas simplesmente não podem ser extraídas. A plataforma Lattes faz um bom trabalho mantendo a estrutura dos currículos uniforme, mas, ainda assim, ocorrem casos como o da Figura 10. Ela mostra um fragmento do resultado da consulta que retorna as áreas de atuação dos professores, assim como o número de docentes em cada uma delas. Como pode ser visto, a ausência de informação não ocorreu uma vez, mas quatorze. No relatório de projetos de pesquisa é possível encontrar diversos projetos que não possuem qualquer tipo de informação sobre seus financiadores, e alguns nem sobre seus integrantes.

Área	Nº de Professores
	14
Administração e Gerência de Redes	1
Análise de Algoritmos e Complexidade de Computação	1

Figura 10 – Resgate de atributo sem valor

Os próximos desafios encontrados se referem à forma como as informações são inseridas nos currículos. O primeiro deles diz respeito aos campos utilizados. Mesmo havendo um local distinto para a inserção cada informação, por vezes os professores não o fazem de forma apropriada. Um exemplo deste caso é no retorno de capítulos de livros publicados, consulta em que é possível encontrar o ano tanto no título do livro quanto no título do capítulo, apesar de haver um campo específico para esta informação. Nos relatórios obtidos, também foi encontrado o nome do financiador de um projeto de pesquisa juntamente

com o título, ao invés da inserção deste no campo apropriado. Apesar destes exemplos não gerarem um grande inconveniente, uma vez que as informações necessárias ainda estão disponíveis, o problema poderia ser maior caso elas estivessem disponíveis apenas na descrição de um projeto de pesquisa, por exemplo, uma vez que o elemento que a contém não é resgatado pelo Ceifador.

A forma como os dados são escritos também pode influenciar os resultados obtidos. A utilização de caracteres especiais, como chaves, por exemplo, não é bem suportada pela Plataforma Lattes. Quando utilizados, o documento XML retirado da plataforma já não os apresenta de forma correta. Como os relatórios utilizam os documentos como fonte de informação, este erro se propaga. Além disso, caso a escrita dos dados não esteja correta, os relatórios gerados podem não representar a realidade. As consultas *XQuery* comparam os conteúdos dos nodos ou atributos. Caso estes tenham sido escritos de forma distinta, a linguagem pode não reconhecer a informação como sendo a mesma. Isto é um problema ainda maior quando levado em conta que sua operação de igualdade considera letras maiúsculas e minúsculas como sendo diferentes. Em relatórios de alunos de orientação concluída foi possível encontrar nomes totalmente escritos em letras maiúsculas, por exemplo. Caso este padrão não seja mantido ao longo do currículo como um todo, *XQuery* não consegue identificá-los como a mesma pessoa quando necessário.

Aldo von Wangenheim	
Professor	Nº de artigos em conjunto
Mario Antonio Ribeiro Dantas	1
Christiane Anneliese Gresse von Wangenheim	
Professor	Nº de artigos em conjunto
Aldo von Wangenheim	1

Figura 11 – Fragmento do relatório de coautorias entre os professores do PPGCC

A Figura 11 mostra outra inconsistência encontrada. Ela mostra um fragmento do relatório gerado para a consulta que retorna, para cada professor, os demais professores do programa com os quais ele mais tem publicações de artigos. Como se pode observar, segundo este relatório a Prof<sup>ca</sup> Christiane tem em seu currículo um artigo publicado em coautoria com o Prof<sup>o</sup> Aldo, mas ele não tem nenhuma publicação com ela. A partir de uma análise detalhada do currículo dele é possível encontrar o nome dela como coautora de dois artigos. Um fragmento

do relatório gerado pela consulta que retorna, para cada professor, os pesquisadores com quem foi coautor em artigos é mostrado na Figura 12. Nela é possível ver o motivo da falha. Em uma ocasião, o nome da professora está incompleto. Em outra, ele está escrito em formato de bibliografia.

Aldo von Wangenheim		
Pesquisador	Artigo	Ano
GRESSE VON WANGENHEIM, CHRISTIANE	AdEQUATE Software Quality Evaluation Model v1.0	2015
Christiane Gresse von Wangenheim	Physical Computing Outreach at Exhibitions	2015

Figura 12 – Fragmento do relatório de coautorias entre todos os professores

O último desafio é uma limitação do sistema desenvolvido. Ele está relacionado ao conteúdo presente nos currículos e à forma como ele é interpretado. Analisando os projetos de pesquisa de cada professor, por exemplo, é visível que alguns deles ocorrem com colaborações internacionais. É possível chegar a esta conclusão observando-se que algumas das informações estão em língua estrangeira. No entanto, é impossível para o banco de dados ou para uma consulta *XQuery* fazer este tipo de análise. Para que este tipo de conclusão seja alcançada, é necessário fazer alguma forma de mineração de dados, o que foge do escopo deste trabalho.



## 5 CONCLUSÕES

O problema inicial deste trabalho era a inexistência de um sistema que auxiliasse a coordenadoria de uma pós-graduação a obter dados e quantitativos que pudessem ser utilizados nas atividades de controle de qualidade, como as avaliações da CAPES ou um planejamento estratégico. Para que um prospectivo sistema fosse aproveitado ao máximo, ele deveria, ainda, permitir que estes relatórios pudessem ser filtradas por período de tempo. Como um dos fatores que motivou este trabalho é acelerar uma atividade anteriormente demorada, não faria sentido desenvolver consultas cujo resultado ainda deveria ser filtrado manualmente pelo usuário. O sistema deveria também ser facilmente implantado em qualquer instituição de ensino que assim desejasse.

A participação da coordenadoria de pós-graduação na definição das consultas garantiu o primeiro requisito. Não há ninguém melhor para saber a quais informações ela gostaria de ter fácil acesso do que a própria.

As necessidades temporais dos resultados, assim como as necessidades de quantitativos foram satisfeitas no desenvolvimento das consultas. Filtros foram aplicados quando necessário e a contabilização dos resultados foi devidamente mostrada ao usuário.

Por fim, considerando-se que a única diferença na utilização do sistema por instituições de ensino distintas consiste nos currículos utilizados, ele pode ser facilmente configurado para ser devidamente utilizado por todas aquelas que desejarem.

Sendo assim, apesar de alguns desafios apresentados no capítulo anterior, considera-se que o objetivo foi atingido. O Ceifador, um sistema que gera relatórios que podem ser filtrados por tempo a partir de currículos XML, foi desenvolvido com sucesso. Ainda assim, os seguintes trabalhos futuros podem aprimorá-lo:

- Ampliar o número de consultas oferecidas;
- Adaptar a interface para que novas consultas sejam facilmente inseridas;
- Ampliar as possibilidades de formatos de exportação dos relatórios obtidos;
- Automatizar o resgate de currículos da plataforma Lattes;

- Adaptar o sistema para ser utilizado em uma versão *web*, permitindo assim que cada professor o acesse e insira seu próprio currículo, acelerando ainda mais o processo;
- Aplicar técnicas de pré e pós-processamento na comparação de informações das consultas desenvolvidas;
- Realizar uma validação dos dados obtidos.

## REFERÊNCIAS

- BASEX. *BaseX Wiki*. 2017.  
 <[http://docs.basex.org/wiki/Main\\_Page](http://docs.basex.org/wiki/Main_Page)>. Acessado em 21/10/2017.
- CAPES. *Capes começa processo de acompanhamento anual da avaliação*. 2006. <<http://www.capes.gov.br/sala-de-imprensa/noticias/1565-blank-53020676>>. Acessado em 21/10/2017.
- CAPES. *Plataforma Sucupira*. 2014.  
 <<http://www.capes.gov.br/avaliacao/plataforma-sucupira>>. Acessado em 21/10/2017.
- CAPES. *APCN - Aplicativo de Propostas de Cursos Novos*. 2015. <<http://www.capes.gov.br/acessoainformacao/perguntas-frequentes/avaliacao-da-pos-graduacao/7409-apcn-aplicativos-de-propostas-de-cursos-novos>>. Acessado em 21/10/2017.
- CNPQ. *Sobre a Plataforma e Histórico*. 2010.  
 <<http://lattes.cnpq.br/>>. Acessado em 21/10/2017.
- FABRO, M. D. D. *SAPos - Sistema de Apoio à Pós-graduação*. 2011. <<http://web.inf.ufpr.br/didonet/sapos-sistema-de-apoio-a-pos-graduacao>>. Acessado em 21/11/2016.
- FAWCETT, J.; AYERS, D.; QUIN, L. R. E. *Beginning XML*. 5. ed. New York, USA: Wiley, 2012. 864 p.
- GNU. *O que é software livre?* 2017.  
 <<http://www.gnu.org/philosophy/free-sw.html>>. Acessado em 21/10/2017.
- GOSLING, J.; MCGILTON, H. *Design Goals for the Java Programming Language*. California, USA: Sun Microsystems Computer Company, 1995. 85 p.
- MICROSOFT. *Compare Typed XML to Untyped XML*. 2016.  
 <<https://msdn.microsoft.com/en-us/library/ms184277.aspx>>. Acessado em 21/10/2017.
- SCOTTINI, A. T. et al. *Siga - sistema de gestão acadêmica para a pós-graduação da ufpr*. 2014.

SOBRINHO, G. N. Coordenação de Aperfeiçoamento de Pessoal de Nível Superior. Dispõe sobre a regulamentação da avaliação quadrienal. Portaria n. 59 de 21 de março, 2017.

UFMS. *Manual do Sistema de Pós-Graduação*. 2013. <<https://sistemas.ufms.br/sigpos/manual.pdf>>. Acessado em 21/10/2017.

UFPEL. *Cobalto Wiki*. 2010. <<http://wikicobalto.ufpel.edu.br/doku.php>>. Acessado em 21/10/2017.

UFPR. *Versão 2.0 do Sistema de Gestão Acadêmica da Pós Graduação (SIGA/UFPR) já está disponível*. 2016. <<http://www.ufpr.br/portalufpr/blog/noticias/versao-2-0-do-sistema-de-gestao-academica-da-pos-graduacao-sigaufpr-ja-esta-disponivel/>>. Acessado em 21/10/2017.

UNICAMP. *Perguntas Frequentes*. 2017. <<https://www.unicamp.br/sipex/help/faq.html?nocache=1479866259304>>. Acessado em 21/10/2017.

W3SCHOOLS. *Learn HTML and CSS with w3schools*. 1. ed. New York, USA: Wiley, 2010. 240 p.

WALMSLEY, P. *XQuery: Search Across a Variety of XML Data*. 2. ed. Massachusetts, USA: O'Reilly Media, 2016. 762 p.

XQJ.NET. *The XQuery API for Java*. 2014. <<http://xqj.net/>>. Acessado em 21/10/2017.



## **APÊNDICE A – Consultas Desenvolvidas**



0. Retorno de nome, data de atualização e caminho dos documentos para visualização e modificação da base de dados:

```
for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$date := $doc/CURRICULO-VITAE/@DATA-ATUALIZACAO
order by $name
return (data($nome), data($date), db:path($doc))
```

1. Lista dos professores:

```
let $c := collection()
return (<tr><th colspan="2" bgcolor="#A7A7A7">Professor</th></tr>,
for $doc in $c
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO
order by $name
return <tr><td colspan="2">{data($name)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total:</td><td>{count($c)}</td></tr>)
```

2. Lista das áreas de atuação dos professores, assim como o número de docentes que atua em cada uma delas:

```
let $a := collection()//AREA-DE-ATUACAO/@NOME-DA-ESPECIALIDADE,
$b := distinct-values(data($a))
return (<tr><th bgcolor="#A7A7A7">Area</th>
<th bgcolor="#A7A7A7">Nº de Professores</th></tr>,
for $i in $b
let $n := count(
for $p in collection()
where $p//AREA-DE-ATUACAO/@NOME-DA-ESPECIALIDADE = $i
return $p)
order by $i
return <tr><td>{data($i)}</td><td>{$n}</td></tr>)
```

3. Para cada professor, os artigos já publicados:

```
for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
("ano início", ..., "ano fim")
return $a)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Artigo</th>
<th bgcolor="#C7C7C7">Ano</th></tr>,
for $a in $c
let $title := $a/DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO,
$year := $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $year, $title
return <tr><td>{data($title)}</td>
```

```

<td>{data($year)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total:</td>
<td>{count($c)}</td></tr>

```

#### 4. Para cada professor, os capítulos de livros já publicados:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $cl in $doc//CAPITULO-DE-LIVRO-PUBLICADO
where $cl/DADOS-BASICOS-DO-CAPITULO/@ANO =
("ano inicio", ..., "ano fim")
return $cl)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Livro</th><th bgcolor="#C7C7C7">Capítulo</th>
<th bgcolor="#C7C7C7">Ano</th></tr>,
for $cl in $c
let $tl := $cl/DETALHAMENTO-DO-CAPITULO/@TITULO-DO-LIVRO,
$tc := $cl/DADOS-BASICOS-DO-CAPITULO/@TITULO-DO-CAPITULO-DO-LIVRO,
$al := $cl/DADOS-BASICOS-DO-CAPITULO/@ANO
order by $al, $tc
return <tr><td>{data($tl)}</td><td>{data($tc)}</td>
<td>{data($al)}</td></tr>,
<tr><td colspan="3" bgcolor="#C7C7C7">Total:</td>
<td>{count($c)}</td></tr>)

```

#### 5. Para cada professor, participantes e financiadores de seus projetos de pesquisa:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c1 := (for $p in $doc//PROJETO-DE-PESQUISA
where $p/@ANO-FIM = ("")
return $p),
$c2 := (for $p in $doc//PROJETO-DE-PESQUISA
where $p/@ANO-FIM = ("ano início", ..., "ano fim")
return $p)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
for $p in $c1
let $pn := $p/@NOME-DO-PROJETO
order by $pn
return (<tr><th bgcolor="#C7C7C7">{data($pn)}</th>
<th bgcolor="#C7C7C7">Em andamento</th></tr>,
for $i in $p//INTEGRANTES-DO-PROJETO/@NOME-COMPLETO
where $i != $name
order by $i
return <tr><td>{data($i)}</td><td>Integrante</td></tr>,
for $f in $p//FINANCIADOR-DO-PROJETO/@NOME-INSTITUICAO
order by $f
return <tr><td>{data($f)}</td><td>Financiador</td></tr>),

```

```

<tr><td bgcolor="#A7A7A7">Total em andamento:</td><td>{count($c1)}</td></tr>,
for $p in $c2
let $pn := $p/@NOME-DO-PROJETO,
$pa := $p/@ANO-FIM
order by $pa, $pn
return (<tr><th bgcolor="#C7C7C7">{data($pn)}</th>
      <th bgcolor="#C7C7C7">{data($pa)}</th></tr>,
for $i in $p//INTEGRANTES-DO-PROJETO/@NOME-COMPLETO
where $i != $name
order by $i
return <tr><td>{data($i)}</td><td>Integrante</td></tr>,
for $f in $p//FINANCIADOR-DO-PROJETO/@NOME-INSTITUICAO
order by $f
return <tr><td>{data($f)}</td><td>Financiador</td></tr>,
<tr><td bgcolor="#A7A7A7">Total concluído:</td><td>{count($c2)}</td></tr>)

```

6. Para cada professor, os periódicos que revisa e já revisou:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c1 := (for $v in $doc//ATUACAO-PROFISSIONAL/VINCULOS
where $v/@OUTRO-VINCULO-INFORMADO = "Revisor de periódico"
and $v/@ANO-FIM = ("")
return $v),
$c2 := (for $v in $doc//ATUACAO-PROFISSIONAL/VINCULOS
where $v/@OUTRO-VINCULO-INFORMADO = "Revisor de periódico"
and $v/@ANO-FIM = ("ano início", ..., "ano fim")
return $v)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th colspan="2" bgcolor="#C7C7C7">Periódico</th>
      <th bgcolor="C7C7C7">Ano de Início</th></tr>,
for $v in $c1
let $pn := $v/../@NOME-INSTITUICAO,
$ai := $v/@ANO-INICIO
order by $ai, $pn
return <tr><td colspan="2">{data($pn)}</td><td>{data($ai)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total em andamento:</td>
      <td>{count($c1)}</td></tr>,
<tr><th bgcolor="#C7C7C7">Periódico</th>
      <th bgcolor="C7C7C7">Ano de Início</th>
      <th bgcolor="C7C7C7">Ano de Fim</th></tr>,
for $v in $c2
let $pn := $v/../@NOME-INSTITUICAO,
$ai := $v/@ANO-INICIO,
$af := $v/@ANO-FIM
order by $af, $ai, $pn
return <tr><td>{data($pn)}</td><td>{data($ai)}</td>
      <td>{data($af)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total concluído:</td>
      <td>{count($c2)}</td></tr>)

```

## 7. Para cada professor, os eventos que organizou:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $e in $doc//DADOS-BASICOS-DA-ORGANIZACAO-DE-EVENTO
where $e/@ANO = ("ano início", ..., "ano fim")
return $e)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Evento</th>
<th bgcolor="#C7C7C7">Ano</th><th bgcolor="#C7C7C7">Tipo</th></tr>,
for $e in $c
let $n := $e/@TITULO,
$a := $e/@ANO,
$t := $e/@TIPO
order by $t, $a, $n
return <tr><td>{data($n)}</td><td>{data($a)}</td>
<td>{data($t)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>
<td>{count($c)}</td></tr>)

```

## 8. Para cada professor, as premiações recebidas:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $p in $doc//PREMIO-TITULO
where $p/@ANO-DA-PREMIACAO = ("ano início", ..., "ano fim")
return $p)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Prêmio</th>
<th bgcolor="#C7C7C7">Ano</th></tr>,
for $p in $c
let $pn := $p/@NOME-DO-PREMIO-OU-TITULO,
$pa := $p/@ANO-DA-PREMIACAO
order by $pa, $pn
return <tr><td>{data($pn)}</td><td>{data($pa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td><td>{count($c)}</td></tr>)

```

## 9. Para cada professor, os alunos de iniciação científica em andamento ou concluída:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c1 := (for $o in $doc//ORIENTACAO-EM-ANDAMENTO-DE-INICIACAO-CIENTIFICA
return $o),
$c2 := (for $o in $doc//OUTRAS-ORIENTACOES-CONCLUIDAS
where $o/DADOS-BASICOS-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/@NATUREZA =
"INICIACAO_CIENTIFICA"
and $o/DADOS-BASICOS-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/@ANO =
("ano início", ..., "ano fim")

```

```

return $o)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th colspan="2" bgcolor="#C7C7C7">IC em andamento</th></tr>,
for $o in $c1
let $no := $o/DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-INICIACAO-
CIENTIFICA/@NOME-DO-ORIENTANDO
order by $no
return <tr><td colspan="2">{data($no)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total em andamento:</td><td>{count($c1)}</td></tr>,
<tr><th bgcolor="#C7C7C7">IC concluída</th>
<th bgcolor="#C7C7C7">Ano</th></tr>,
for $o in $c2
let $no := $o/DETALHAMENTO-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/
@NOME-DO-ORIENTADO,
$sao := $o/DADOS-BASICOS-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/@ANO
order by $ao, $no
return <tr><td>{data($no)}</td><td>{data($ao)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total concluído:</td><td>{count($c2)}</td></tr>)

```

10. Para cada professor, os alunos que foram seus orientandos de TCC e também de mestrado ou doutorado:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$g := $doc//OUTRAS-ORIENTACOES-CONCLUIDAS,
$gn := $g/DETALHAMENTO-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/
@NOME-DO-ORIENTADO,
$c1 := (for $ma in $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-
MESTRADO/@NOME-DO-ORIENTANDO
where $ma = $gn
return $ma),
$c2 := (for $mc in $doc//ORIENTACOES-CONCLUIDAS-PARA-MESTRADO
where $mc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/
@NOME-DO-ORIENTADO = $gn
and $mc/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@ANO =
("ano início", ..., "ano fim")
return $mc),
$c3 := (for $da in $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-
DOCTORADO/@NOME-DO-ORIENTANDO
where $da = $gn
return $da),
$c4 := (for $dc in $doc//ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO
where $dc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO/
@NOME-DO-ORIENTADO = $gn
and $dc/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO/@ANO =
("ano início", ..., "ano fim")
return $dc)
where $g/DADOS-BASICOS-DE-OUTRAS-ORIENTACOES-CONCLUIDAS/@NATUREZA =
"TRABALHO_DE_CONCLUSAO_DE_CURSO_GRADUACAO"
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,

```

```

<tr><th colspan="2" bgcolor="#C7C7C7">Mestrado em andamento</th></tr>,
for $ma in $c1
order by $ma
return <tr><td>{data($ma)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total em andamento:</td><td>{count($c1)}</td></tr>,
<tr><th bgcolor="#C7C7C7">Mestrado Concluído</th>
  <th bgcolor="#C7C7C7">Ano</th></tr>,
for $mc in $c2
let $mcn := $mc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/
@NOME-DO-ORIENTADO
let $mca := $mc/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/
@ANO
order by $mca, $mcn
return <tr><td>{data($mcn)}</td><td>{data($mca)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total concluído:</td><td>{count($c2)}</td></tr>,
<tr><th colspan="2" bgcolor="#C7C7C7">Doutorado em andamento</th></tr>,
for $da in $c3
order by $da
return <tr><td>{data($da)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total em andamento:</td><td>{count($c3)}</td></tr>,
<tr><th bgcolor="#C7C7C7">Doutorado Concluído</th>
  <th bgcolor="#C7C7C7">Ano</th></tr>,
for $dc in $c4
let $dcn := $dc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-DOUTORADO/
@NOME-DO-ORIENTADO
let $dca := $dc/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-DOUTORADO/
@ANO
order by $dca, $dcn
return <tr><td>{data($dcn)}</td><td>{data($dca)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total concluído:</td><td>{count($c4)}</td></tr>)

```

11. Para cada professor, os alunos que concluíram o mestrado sob sua orientação:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $m in $doc//ORIENTACOES-CONCLUIDAS-PARA-MESTRADO
(:where $m/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@ANO =
("ano início", ..., "ano fim")):)
return $m)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Mestre</th>
  <th bgcolor="#C7C7C7">Ano</th></tr>,
for $m in $c
let $mn := $m/DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/
@NOME-DO-ORIENTADO,
$ma := $m/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@ANO
order by $ma, $mn
return <tr><td>{data($mn)}</td><td>{data($ma)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total:</td><td>{count($c)}</td></tr>)

```



12. Para cada professor, os alunos de mestrado coautores em artigos:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c1 := (for $ma in $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-
MESTRADO/@NOME-DO-ORIENTANDO
return for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO = ("ano início", ...,
"ano fim") and $a/AUTORES/@NOME-COMPLETO-DO-AUTOR = $ma
return $a),
$c2 := (for $mc in $doc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-
MESTRADO/@NOME-DO-ORIENTADO
return for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO = ("ano início", ...,
"ano fim") and $a/AUTORES/@NOME-COMPLETO-DO-AUTOR = $mc
return $a)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Mestrando</th>
<th bgcolor="C7C7C7">Artigo</th><th bgcolor="C7C7C7">Ano</th></tr>,
for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
("ano início", ..., "ano fim")
let $tit := $a/DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO
let $aa := $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $aa, $tit
return for $au in $a/AUTORES/@NOME-COMPLETO-DO-AUTOR
let $ma := $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-MESTRADO/
@NOME-DO-ORIENTANDO
where $au = $ma
order by $au
return <tr><td>{data($au)}</td><td>{data($tit)}</td>
<td>{data($aa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>
<td>{count($c1)}</td></tr>,
<tr><th bgcolor="C7C7C7">Mestre</th>
<th bgcolor="C7C7C7">Artigo</th><th bgcolor="C7C7C7">Ano</th></tr>,
for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
("ano início", ..., "ano fim")
let $tit := $a/DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO
let $aa := $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $aa, $tit
return for $au in $a/AUTORES/@NOME-COMPLETO-DO-AUTOR
let $mc := $doc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/
@NOME-DO-ORIENTADO
where $au = $mc
order by $au
return <tr><td>{data($au)}</td><td>{data($tit)}</td>
<td>{data($aa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>

```

```
<td>{count($c2)}</td></tr>
```

13. Para cada professor, os alunos que concluíram o doutorado sob sua orientação:

```
for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $d in $doc//ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO
where $d/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO/@ANO =
("ano início", ..., "ano fim")
return $d)
order by $name
return (<tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Doutor</th>
<th bgcolor="#C7C7C7">Ano</th></tr>,
for $d in $c
let $dn := $d/DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO/
@NOME-DO-ORIENTADO,
$da := $d/DADOS-BASICOS-DE-ORIENTACOES-CONCLUIDAS-PARA-DOCTORADO/@ANO
order by $da, $dn
return <tr><td>{data($dn)}</td><td>{data($da)}</td></tr>,
<tr><td bgcolor="#C7C7C7">Total:</td><td>{count($c)}</td></tr>)
```

14. Para cada professor, os alunos de doutorado coautores em artigos:

```
for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c1 := (for $da in $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-
DOCTORADO/@NOME-DO-ORIENTANDO
return for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO = ("ano início", ...,
"ano fim") and $a/AUTORES/@NOME-COMPLETO-DO-AUTOR = $da
return $a),
$c2 := (for $dc in $doc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-
DOCTORADO/@NOME-DO-ORIENTADO
return for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO = ("ano início", ...,
"ano fim") and $a/AUTORES/@NOME-COMPLETO-DO-AUTOR = $dc
return $a)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Doutorando</th>
<th bgcolor="#C7C7C7">Artigo</th><th bgcolor="#C7C7C7">Ano</th></tr>,
for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
("ano início", ..., "ano fim")
let $tit := $a/DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO
let $aa := $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $aa, $tit
return for $au in $a/AUTORES/@NOME-COMPLETO-DO-AUTOR
```

```

let $da := $doc//DETALHAMENTO-DA-ORIENTACAO-EM-ANDAMENTO-DE-DOUTORADO/
@NOME-DO-ORIENTANDO
where $au = $da
order by $au
return <tr><td>{data($au)}</td><td>{data($tit)}</td>
      <td>{data($aa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>
      <td>{count($c1)}</td></tr>,
<tr><th bgcolor="#C7C7C7">Doutor</th>
      <th bgcolor="C7C7C7">Artigo</th><th bgcolor="C7C7C7">Ano</th></tr>,
for $a in $doc//ARTIGO-PUBLICADO
where $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
      ("ano início", ..., "ano fim")
let $tit := $a/DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO
let $aa := $a/DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $aa, $tit
return for $au in $a/AUTORES/@NOME-COMPLETO-DO-AUTOR
      let $dc := $doc//DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-
      DOUTORADO/@NOME-DO-ORIENTADO
      where $au = $dc
      order by $au
      return <tr><td>{data($au)}</td><td>{data($tit)}</td>
            <td>{data($aa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>
      <td>{count($c2)}</td></tr>

```

15. Para cada professor, os pesquisadores com quem foi coautor em artigos num dado intervalo de tempo:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $au in $doc//ARTIGO-PUBLICADO/AUTORES
where $au/../DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO =
      ("ano início", ..., "ano fim")
and $au/@NOME-COMPLETO-DO-AUTOR != $name
return $au)
order by $name
return (<tr><th colspan="3" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Pesquisador</th>
      <th bgcolor="#C7C7C7">Artigo</th>
      <th bgcolor="#C7C7C7">Ano</th></tr>,
for $au in $c
let $an := $au/@NOME-COMPLETO-DO-AUTOR,
$tit := $au/../DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO,
$aa := $au/../DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO
order by $aa, $tit, $an
return <tr><td>{data($an)}</td><td>{data($tit)}</td>
      <td>{data($aa)}</td></tr>,
<tr><td colspan="2" bgcolor="#C7C7C7">Total:</td>
      <td bgcolor="#C7C7C7">{count($c)}</td></tr>)

```

16. Para cada professor, os demais professores do mesmo programa com quem colabora em projetos de pesquisa:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO,
$nn := collection()//DADOS-GERAIS/@NOME-COMPLETO,
$c := (for $p in $doc//PROJETO-DE-PESQUISA
for $i in $p//INTEGRANTES-DO-PROJETO
where $p/@ANO-FIM = ("")
and $i/@NOME-COMPLETO = $nn
and $i/@NOME-COMPLETO != $name
return $i),
$c1 := (for $p in $doc//PROJETO-DE-PESQUISA
for $i in $p//INTEGRANTES-DO-PROJETO
where $p/@ANO-FIM = ("ano inicio", ..., "ano fim")
and $i/@NOME-COMPLETO = $nn
and $i/@NOME-COMPLETO != $name
return $i)
order by $name
return <tr><th colspan="4" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Pesquisador</th>
<th colspan="2" bgcolor="#C7C7C7">Projeto</th>
<th bgcolor="#C7C7C7">Ano de Início</th></tr>,
for $i in $c
let $in := $i/@NOME-COMPLETO,
$p := $i/../../@NOME-DO-PROJETO,
$a := $i/../../@ANO-INICIO
order by $a, $p, $in
return <tr><td>{data($in)}</td><td colspan="2">{data($p)}</td>
<td>{data($a)}</td></tr>,
<tr><td colspan="3" bgcolor="#C7C7C7">Total em andamento:</td>
<td bgcolor="#C7C7C7">{count($c)}</td></tr>,
<tr><th bgcolor="#C7C7C7">Pesquisador</th><th bgcolor="#C7C7C7">Projeto</th>
<th bgcolor="#C7C7C7">Ano de Início</th>
<th bgcolor="#C7C7C7">Ano de Fim</th></tr>,
for $i in $c1
let $in := $i/@NOME-COMPLETO,
$p := $i/../../@NOME-DO-PROJETO,
$ai := $i/../../@ANO-INICIO,
$af := $i/../../@ANO-FIM
order by $af, $ai, $p, $in
return <tr><td>{data($in)}</td><td>{data($p)}</td><td>{data($ai)}</td>
<td>{data($af)}</td></tr>,
<tr><td colspan="3" bgcolor="#C7C7C7">Total concluído:</td>
<td bgcolor="#C7C7C7">{count($c1)}</td></tr>

```

17. Para cada professor, os N professores do mesmo programa com quem mais publicou artigos, assim como o número de artigos publicados em conjunto:

```

for $doc in collection()
let $name := $doc//DADOS-GERAIS/@NOME-COMPLETO
order by $name
return <tr><th colspan="2" bgcolor="#A7A7A7">{data($name)}</th></tr>,
<tr><th bgcolor="#C7C7C7">Professor</th>
<th bgcolor="#C7C7C7">Nº de artigos em conjunto</th></tr>,

```

```

for $i in 1 to N
return (for $p in collection()
let $pn := $p//DADOS-GERAIS/@NOME-COMPLETO,
$art := $doc//ARTIGO-PUBLICADO
where $p != $doc and $pn = $art/AUTORES/@NOME-COMPLETO-DO-AUTOR
let $n := count(for $x in $art/AUTORES/@NOME-COMPLETO-DO-AUTOR
where $x = $pn
return $x)
order by $n descending
return <tr><td>{data($pn)}</td><td>{data($n)}</td></tr>)[$i])

```



## **ANEXO A – Avaliação PPGCC UFSC 2013**





## Ficha de Avaliação do Programa

**Período de Avaliação:** 2010 a 2012 **Etapa:** Avaliação Trienal 2013  
**Área de Avaliação:** 2 - CIÊNCIA DA COMPUTAÇÃO  
**IES:** 41001010 - UFSC - UNIVERSIDADE FEDERAL DE SANTA CATARINA  
**Programa:** 41001010025P2 - CIÊNCIAS DA COMPUTAÇÃO  
**Modalidade:** Acadêmico

Curso	Nível	Ano Início	Ano Início
CIÊNCIAS DA COMPUTAÇÃO	Mestrado	1992	
Ciência da Computação	Doutorado		2012

### Dados Disponíveis na Coleta de Dados

Curso	Nível	Ano	Ano	Ano
CIÊNCIAS DA COMPUTAÇÃO	Mestrado	2010	2011	2012
Ciência da Computação	Doutorado			2012

## 1 - PROPOSTA DO PROGRAMA

Itens de Avaliação	Peso	Avaliação
1.1. Coerência, consistência, abrangência e atualização das áreas de concentração, linhas de pesquisa, projetos em andamento e proposta curricular.	35.00	Bom
1.2. Planejamento do programa com vistas a seu desenvolvimento futuro, contemplando os desafios internacionais da área na produção do conhecimento, seus propósitos na melhor formação de seus alunos, suas metas quanto à inserção social mais rica dos seus egressos, conforme os parâmetros da área.	35.00	Bom
1.3. Infraestrutura para ensino, pesquisa e, se for o caso, extensão.	30.00	Bom

**Comissão:****Bom**

### Apreciação

A proposta do programa atende aos requisitos definidos no documento de área no que tange aos critérios de coerência e abrangência das áreas de concentração. Há, contudo, um desbalanceamento entre as áreas de concentração. A área de Sistemas de Conhecimento possui apenas uma linha de pesquisa, enquanto que a linha de Sistemas de Computação possui 5 linhas de pesquisa.

Tanto o Mestrado quanto o Doutorado possuem 4 disciplinas obrigatórias, duas delas cobrindo áreas núcleo da Ciência da Computação, como Algoritmos e Teoria da Computação, além de um leque amplo de disciplinas que cobrem as áreas de atuação do programa. Os alunos são estimulados a cursar uma disciplina de Métodos Estatísticos e Projeto de Experimentos. A proposta do programa atende aos requisitos definidos no documento de área no que tange aos critérios de coerência e abrangência das áreas de concentração.

O programa tem buscado colaborações com instituições de boa reputação acadêmica no exterior, incluindo Estados Unidos, Canadá e Europa, e também no Brasil.

A infraestrutura de ensino e pesquisa é boa, com 14 laboratórios (alguns muito bem equipados), permitindo o desenvolvimento de atividades de pós-graduação em bom nível.

## 2 - CORPO DOCENTE

Itens de Avaliação	Peso	Avaliação
2.1. Perfil do corpo docente, consideradas titulação, diversificação na origem de formação, aprimoramento e experiência, e sua compatibilidade e adequação à Proposta do Programa.	30.00	Muito Bom

## Ficha de Avaliação do Programa

2.2. Adequação e dedicação dos docentes permanentes em relação às atividades de pesquisa e de formação do programa.	30.00	Bom
2.3. Distribuição das atividades de pesquisa e de formação entre os docentes do programa.	30.00	Bom
2.4. Contribuição dos docentes para atividades de ensino e/ou de pesquisa na graduação, com atenção tanto à repercussão que este item pode ter na formação de futuros ingressantes na PG, quanto (conforme a área) na formação de profissionais mais capacitados no plano da graduação.	10.00	Regular

**Comissão:** Bom

### Apreciação

O corpo docente tem boa formação, sendo que vários docentes concluíram sua formação em instituições de alto nível em Ciência da Computação no Brasil e no exterior, com alguns docentes com formação em Engenharia Elétrica e em Engenharia de Produção.

Com relação ao percentual de docentes com bolsa de produtividade em pesquisa do CNPq, houve um avanço com relação ao triênio anterior. O programa possui também 2 bolsistas de produtividade em desenvolvimento tecnológico e inovação. Neste quesito, o programa está situado entre os 30% melhores da área.

Vários docentes têm participado de comitês científicos de eventos nacionais e no exterior, bem como na revisão de artigos de diversos periódicos internacionais de bom nível.

O programa apresenta um número expressivo de projetos de pesquisa em Ciência da Computação de alto nível, financiados por agências de fomento à pesquisa. A distribuição da coordenação de tais projetos foi irregular, ficando concentrada em apenas um terço do corpo docente.

Com relação às demais atividades, a dedicação do corpo docente foi regular. Embora a maioria dos docentes atue em dedicação exclusiva no programa, 9 dos 30 docentes permanentes (ou seja, cerca de 30%) atuaram também em outro programa. Este número é considerado elevado em se tratando de um programa que está iniciando seu doutorado.

Com respeito à atuação dos docentes na graduação, o relatório aponta que os docentes participam de orientações de iniciação científica e de trabalhos de conclusão de cursos, bem como da supervisão de alunos da pós-graduação que atuam em estágio de docência na graduação.

### 3 - CORPO DISCENTE, TESES E DISSERTAÇÕES

Itens de Avaliação	Peso	Avaliação
3.1. Quantidade de teses e dissertações defendidas no período de avaliação, em relação ao corpo docente permanente e à dimensão do corpo discente.	35.00	Muito Bom
3.2. Distribuição das orientações das teses e dissertações defendidas no período de avaliação em relação aos docentes do programa.	15.00	Bom
3.3. Qualidade das Teses e Dissertações e da produção de discentes autores da pós-graduação e da graduação (no caso de IES com curso de graduação na área) na produção científica do programa, aferida por publicações e outros indicadores pertinentes à área.	40.00	Bom
3.4. Eficiência do Programa na formação de mestres e doutores bolsistas: Tempo de formação de mestres e doutores e percentual de bolsistas titulados.	10.00	Bom

**Comissão:** Bom

### Apreciação

O programa formou 97 mestres no triênio, um número significativo, acima da média nacional da área.

A maioria significativa dos docentes teve orientações de dissertações concluídas durante o triênio. A distribuição do número de orientações apresenta ligeiro desbalanceamento.

## Ficha de Avaliação do Programa

O programa mantém cooperações com pesquisadores e grupos de outras instituições de bom nível tanto no Brasil quanto no exterior.

Os discentes e egressos têm boa participação na produção do programa. Durante o triênio, os discentes e egressos participaram como coautores de 171 publicações, próximo à média dos programas de mesmo nível. O programa possui uma comissão que verifica e habilita os discentes para a defesa, conforme suas publicações. O relatório reporta que 5 artigos receberam premiações em eventos, sendo 1 em evento internacional.

Os alunos concluem as dissertações em tempo razoável (entre 30 e 36 meses) tendo em vista os parâmetros da área e considerando sua participação e envolvimento nas publicações.

O doutorado foi criado em 2012 e esse quesito não se aplica.

### 4 - PRODUÇÃO INTELECTUAL

Itens de Avaliação	Peso	Avaliação
4.1. Publicações qualificadas do Programa por docente permanente.	65.00	Bom
4.2. Distribuição de publicações qualificadas em relação ao corpo docente permanente do Programa.	30.00	Muito Bom
4.3. Produção técnica, patentes e outras produções consideradas relevantes.	5.00	Bom
Comissão:		Bom

#### Apreciação

A produção acadêmica qualificada no triênio foi boa, evidenciada pelo índice geral, que ficou entre os 50% mais produtivos da área, compatível com os programas de nível 4. Dos 33 docentes reportados, 27 (ou seja, cerca de 80%) tiveram pelo menos uma publicação no estrato restrito (conferências e periódicos A1-B1), e destes, 15 tiveram pelo menos uma publicação em periódicos A1-B1 no triênio. As publicações qualificadas têm boa participação de autores discentes ou de egressos.

O programa apresenta boa produção tecnológica, incluindo projetos de inovação com empresas e o desenvolvimento de ferramentas de software.

### 5 - INSERÇÃO SOCIAL

Itens de Avaliação	Peso	Avaliação
5.1. Inserção e impacto regional e (ou) nacional do programa.	35.00	Muito Bom
5.2. Integração e cooperação com outros programas e centros de pesquisa e desenvolvimento profissional relacionados à área de conhecimento do programa, com vistas ao desenvolvimento da pesquisa e da pós-graduação.	30.00	Muito Bom
5.3 - Visibilidade ou transparência dada pelo programa à sua atuação.	15.00	Bom
5.4. Internacionalização.	20.00	Bom
Comissão:		Muito Bom

#### Apreciação

O programa tem bom impacto regional, com uma formação significativa de mestres no triênio. O programa mantém cooperação com universidades no exterior, com Universidades da Alemanha, Inglaterra, Itália, Estados Unidos, Canadá e França.

A instituição criou há dez anos um centro para geração de negócios e empreendimentos de base tecnológica em TI, que tem estimulado a geração de empresas. Além disso, o programa mantém diversos projetos de extensão com empresas, com participação de docentes.

O programa disponibiliza informações sobre suas atividades acadêmicas em páginas atualizadas e com

## Ficha de Avaliação do Programa

elogiável transparência.

### Qualidade dos Dados

Quesitos	Qualidade
1 - PROPOSTA DO PROGRAMA	Regular
2 - CORPO DOCENTE	Regular
3 - CORPO DISCENTE, TESES E DISSERTAÇÕES	Bom
4 - PRODUÇÃO INTELECTUAL	Bom
5 - INSERÇÃO SOCIAL	Regular
Comissão:	
Bom	

### Comentário

Em 2010 e 2012 o relatório aponta que nenhum dos docentes do programa ministrou aulas na graduação. Não fica claro se de fato os docentes não atuaram ou se houve erro no preenchimento.

### Conceito/Nota CA

Quesitos	Peso	Avaliação Comissão
1 - PROPOSTA DO PROGRAMA	0.00	Bom
2 - CORPO DOCENTE	20.00	Bom
3 - CORPO DISCENTE, TESES E DISSERTAÇÕES	30.00	Bom
4 - PRODUÇÃO INTELECTUAL	40.00	Bom
5 - INSERÇÃO SOCIAL	10.00	Muito Bom
<b>Data Chancela:</b> 28/11/2013		<b>Conceito Comissão:</b> Bom
		<b>Nota Comissão:</b> 4

### Apreciação

O programa passou por mudanças recentes, principalmente em função da criação do doutorado, em 2012. Entre as mudanças estão a reestruturação de disciplinas e carga horária e a definição de metas mais rígidas para credenciamento e permanência de docentes no programa.

A produção acadêmica qualificada é compatível com a dos programas de nível 4.

Para a consolidação do doutorado, é importante incrementar a produção intelectual do programa em veículos de boa qualidade.

### Complementos

#### Apreciações ou sugestões complementares sobre a situação ou desempenho do programa.

#### Recomendações da Comissão ao Programa.

Recomenda-se moderação na captação de alunos de doutorado, bem como incentivar o desenvolvimento de teses de alta qualidade. O relatório reporta que 9 dos 30 docentes permanentes (ou seja, cerca de 30%) atuaram também em outro programa. Apesar desse percentual estar no limite do que é recomendado, recomenda-se moderação na atuação do corpo docente em outros programas, priorizando a consolidação do novo doutorado.

**A CAPES deve promover visita de consultores ao Programa?** Não

**Justificativa da recomendação de visita ao programa.**

**A Comissão recomenda mudança de área de avaliação?** Não

**Área Indicada:**

**Justificativa da recomendação de mudança de área de avaliação do programa (em caso afirmativo)**

### Nota CTC-ES

<b>Data Chancela:</b>	<b>Nota CTC-ES:</b> 4
-----------------------	-----------------------

### Apreciação

O CTC-ES, na 150ª reunião, ratificou a análise e a nota atribuída pela Comissão de Área ao presente programa.

<b>Comissão Responsável pela Avaliação:</b>	<b>Sigla IES</b>
ADENAUER CORREA YAMIN	UFPEL Consultor(a)

## Ficha de Avaliação do Programa

Comissão Responsável pela Avaliação:	Sigla IES	
ALBA CRISTINA MAGALHAES ALVES DE MELO	UNB	Consultor(a)
ALBERTO FERREIRA DE SOUZA	UFES	Consultor(a)
ALBERTO HENRIQUE FRADE LAENDER	UFMG	Consultor(a)
ALTIGRAN SOARES DA SILVA	UFAM	Coordenador(a) Adjunto(a)
ARTUR ZIVIANI	LNCC	Consultor(a)
AUGUSTO CESAR ALVES SAMPAIO	UFPE	Consultor(a)
CELIO VINICIUS NEVES DE ALBUQUERQUE	UFF	Consultor(a)
CLAUDIA LINHARES SALES	UFC	Consultor(a)
CLÁUDIO LEONARDO LUCCHESI	UFMS	Consultor(a)
EDSON NORBERTO CÁCERES	UFMS	Consultor(a)
HERMES SINGER	UFSCAR	Consultor(a)
JAYME LUIZ SZWARCFITER	UFRJ	Consultor(a)
LUIS DA CUNHA LAMB	UFRGS	Consultor(a)
MARCO ANTONIO CASANOVA	PUC-RIO	Consultor(a)
MARTA LIMA DE QUEIROS MATTOSO	UFRJ	Consultor(a)
NELSON LUIS SALDANHA DA FONSECA	UNICAMP	Consultor(a)
PAULO CESAR MASIERO	USP/SC	Consultor(a)
PAULO HENRIQUE LEMELLE FERNANDES	PUC/RS	Consultor(a)
PHILIPPE OLIVIER ALEXANDRE NAVAUUX	UFRGS	Coordenador(a)
RODOLFO JARDIM DE AZEVEDO	UNICAMP	Coordenador(a) Adjunto(a) Mestrado Profissional
SANDRA APARECIDA DE AMO	UFU	Consultor(a)
SIANG WUN SONG	UFABC	Consultor(a)
THAIS VASCONCELOS BATISTA	UFRN	Consultor(a)